ANNÉE UNIVERSITAIRE 1983-1984

PREMIER & DEUXIÈME CYCLES

CENTRE
DE TELE
ENSEIGNEMENT

COURS
PAR
CORRESPONDANCE

UNIVERSITE DE PROVENCE centre d'aix
aix marseille 1
29, AVENUE ROBERT SCHUMAN - BUREAU N° 63

MONSIEUR PETIT

COURS PAR CORRESPONDANCE N° 1

"INITIATION À L'INFORMATIQUE"

MAS 771/2 UC

1983-1984

CENTRE DE CALCUL DE L'UNIVERSITE DE PROVENCE

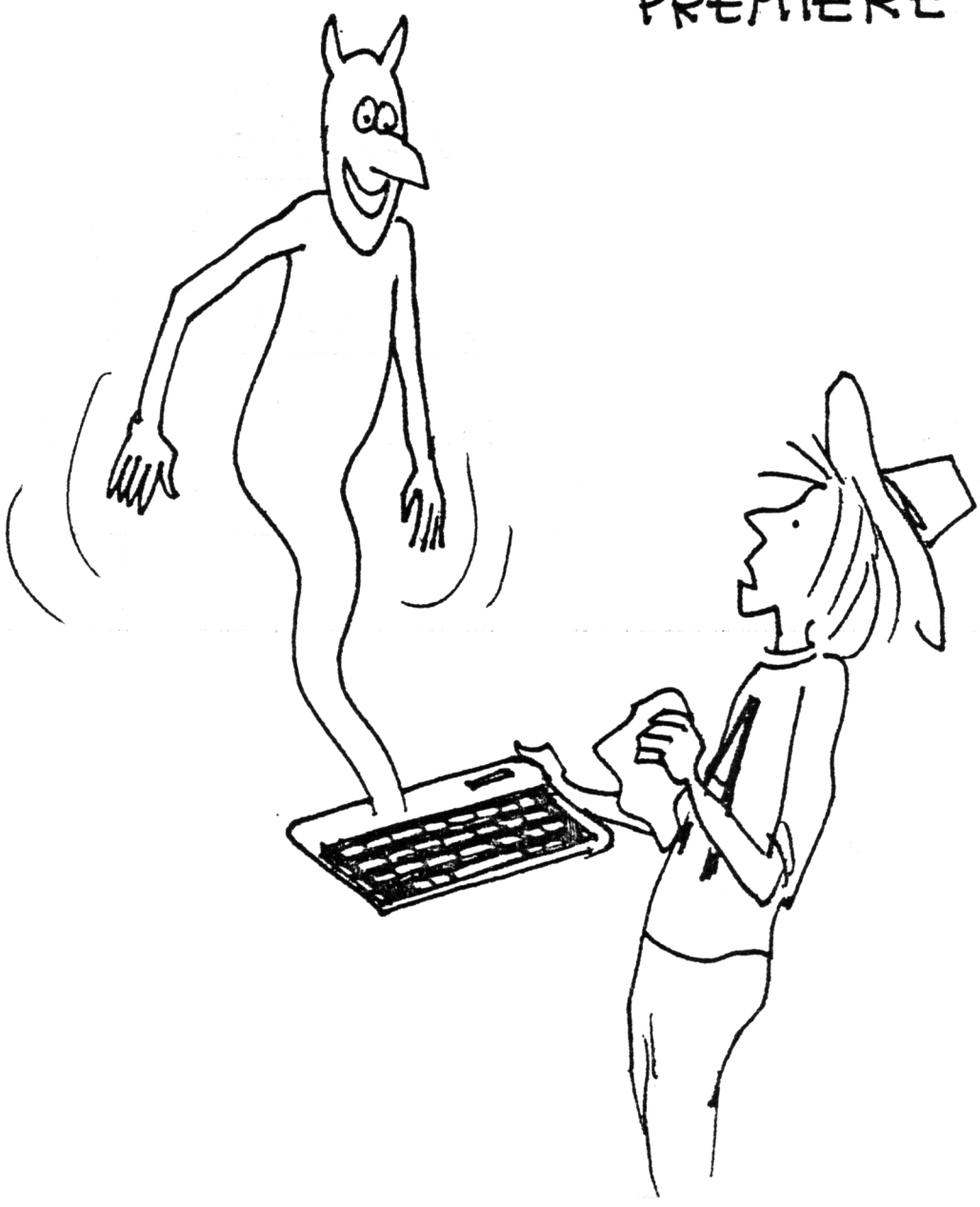
SERVICE COMMUN

INITIATION AU BASIC Niveau I

Par
Jean-Pierre PETIT
Maitre de Recherche au CNRS
Direction adjoint du Centre de Calcul
de l'Université d'Aix-Marseille.

Matériel conseillé : ORIC-1 48 K

PREMIÈRE PARTIE



INTRODUCTION :

L'informatique est devenue une véritable mode. Pourquoi ce phénomène est-il devenu soudain important dans nos sociétés industrialisées ?

Il existe dans la région une usine de production de "puces". Elle est à Rousset et emploie déja plusieurs centaines de personnes. C'est un monde étrange. Dans cette provence si souriante, vous trouverez un grand bâtiment rectangulaire, sans fenêtre.

On dirait une ruche plate. Avec une toute petite entrée. Ce qui rentre dans cette ruche, ce sont des cylindres de Silicium. Des monocristaux. Sous cette forme le Silicium ressemble à du verre. Il est débité en fins disques, d'une dizaine de centimètres de diamètre, et épaisses d'une fraction de millimètre. Sur ces galettes, au prix d'opérations très sophistiquées, on va déposer une douzaine de couches, alternances de conducteurs et d'isolants. Cet ensemble va lentement constituer soit un bloc mémoire, soit un microprocesseur.

Des produits chimiques variés seront utilisés pour "doper" ces galettes de Silicium. Il y a de tout, y compris des produits hautement toxiques qu'il faut manipuler avec les plus grandes précautions.

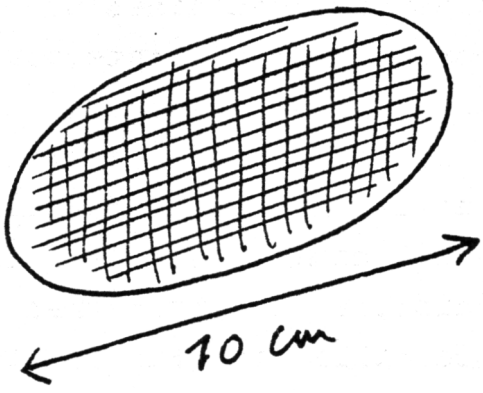
C'est là que se fabrique le vingt et unième siècle. Les travailleurs de cette usine ont d'ailleurs de drôles d'allures. Ils ont des bottes, des blouses, des gants de caoutchouc, des cagoules. Les opérations les plus délicates se font dans des "salles blanches". C'est à dire des salles où règne une propreté absolue. Les reliefs créés font en effet moins du micron. Il n'est donc pas question de laisser de minuscules poussières se déposer sur les disques pendant les opérations.

On pénètre dans ces salles blanches après avoir subi une douche aux ultra sons.

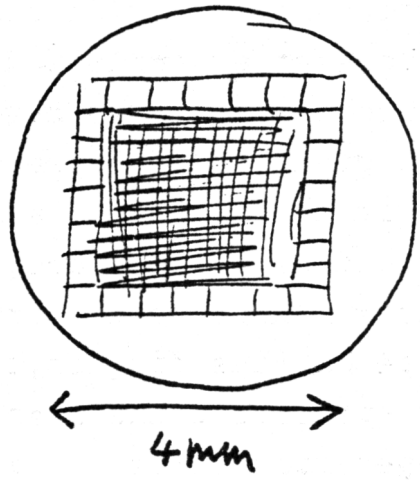
Quand les "puces" sont fabriquées, on les teste. Sur des bancs spéciaux. Il y a plusieurs centaines de puces sur chaque galette. Dans l'opération de test de minuscules aiguilles métalliques descendent et vont griffer des petites zônes prévues à cet effet. Les circuits sont ainsi testé, les uns après les autres.

Quarante pour cent de ces puces seront reconnues défectueuses et marquées, à l'aide d'un point d'encre rouge.

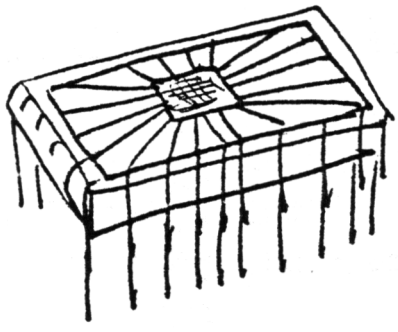
Un peu plus loin les galettes seront sciées et les petites Puces ,qui ne font qu'une dizaine de millimètre carrés chacune, vont être collées sur des pavés de céramique noire. La puce reçoit alors ses "pattes".

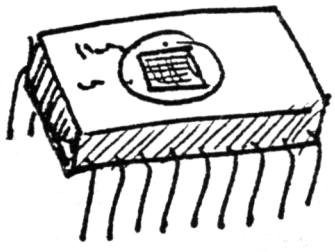


Galette de Silicium



Une puce sous le microscope.
A l'extérieur les carrés de matière conductrice sur lesquels s'appuieront les aiguilles test.





La puce est reliée à ses pattes. Une fois le montage achevée sur certaines puces on peut encore apercevoir le petit carré de Silicium à travers unepetite fenêtre de quartz.

Ce sont ces petits objets qui vont transformer nos vies.

Ils vont devenir microprocesseurs,mémoires.
Prix de revient à l'unité : QUELQUES FRANCS !

Tout est là.L'informatique n'est pas chère.Elle ne nécéssite pratiquement aucune matière première,elle ne consomme presque pas d'énergie.

La révolution informatique va donc à contre courant vis à vis de l'évolution technologique classique.C'est pour cette raison que rien ne pourra l'arrêter.

Avec ces puces on va fabriquer des ordinateurs.Des micro ordinateurs.Tout cela a nécéssité des investissements très importants.Mais le projet APPOLO en a supporté la plus grande part.C'est une retombée de cette conquête de la Lune.Tellement importante que chaque dollar dépensé pour mettre le pied sur notre satellite a déja rapporté aux pays hautement industrialisés de deux à trois dollars.

Cette industrie du micro ordinateur se développe comme une trainée de poudre.

Qu'est-ce que c'est,un micro ordinateur ?
Un boitier en plastique,une alimentation,un clavier.Un circuit imprimé qui recevra les "chips".Tout cela va être relié à une "visu",qui peut être une télévision du commerce,équipée d'une prise "PERITEL" qui est en train de devenir un standard.

Les premiers micro ordinateurs largement diffusés furent les célèbres APPLE II,dont les performances sont telles qu'ils occuppent encore (pour un temps) le haut du pavé,offrant le meilleur rapport qualité prix.

Il y a cinq ans ces machines pouvaient être acquises pour un prix se situant aux alentours de 15 000 F.

Ajourd'hui ce prix est tombé à 7000 F.Où cela s'arrètera-t-il ?
Et d'abord,combien ça côute,un APPLE II en "sortie d'usine" ?

540 F !!!!

Inutile de dire que cette opération a été hautement profitable.
Il ya cinq ans on vendait la rangée de six chips constituant un "bloc mémoire" près de sept cent francs.Pas mal pour des objets qui,au prix de fabrication,côutaient moins de trente francs !

L'évolution future est donc évidente : Les prix vont baisser, baisser.

Ce cours est axé sur un matériel anglais : l'ORIC-1. Ces petites machines sont presque aussi performantes que les célèbres APPLE II en dépit de leur présentation rustique. Elles onf la même capacité mémoire : 48 K utilisateur. Et elles valent à peine plus de 2000 F. Aujourd'hui....

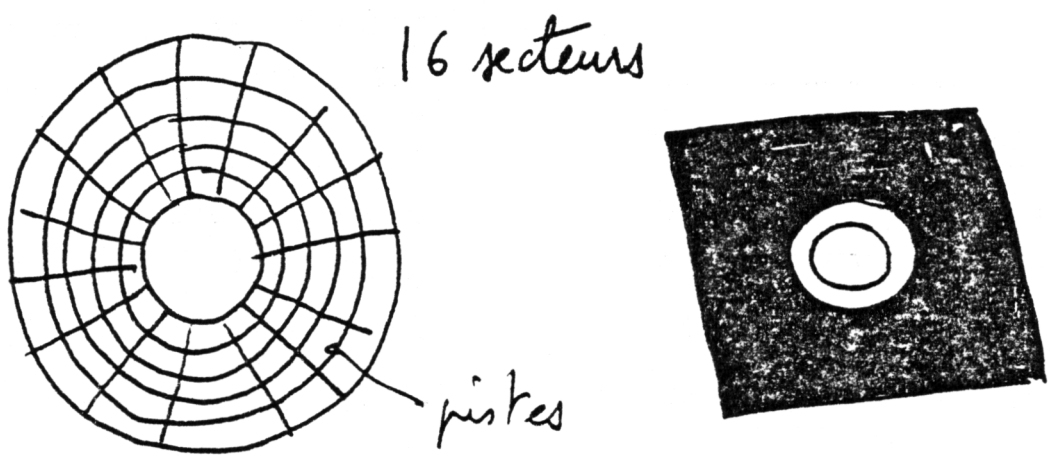
Nous n'avons pas encore de matériels comparables en France. Tout simplement parce que les français n'avaient pas compris l'importance du marché. Les chaines de production de Thomson avaient été au départ dimensionnées pour 3000 machines. Celles de l'ORIC pour 300 000 !!!

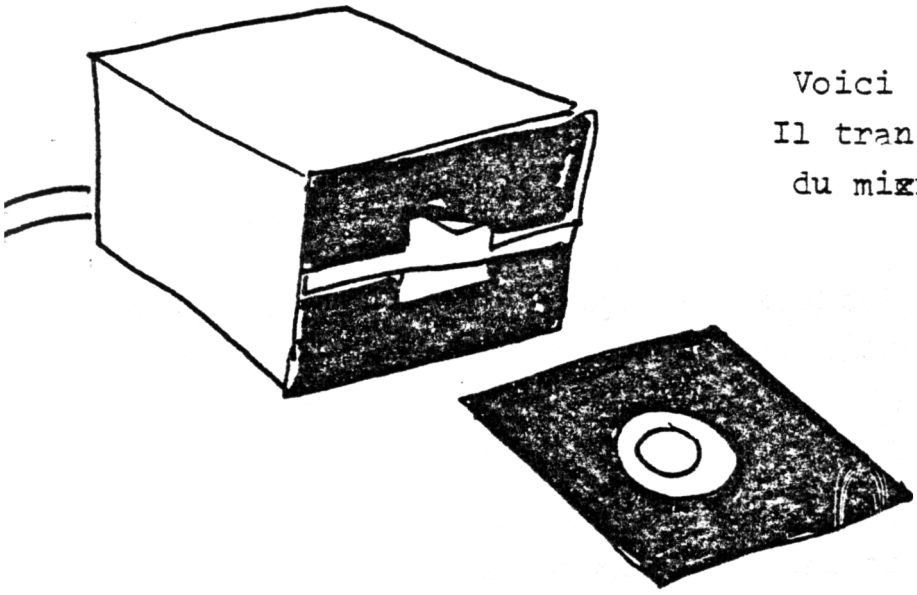
Ces micro ordinateurs doivent être reliés à une mémoire externe. En effet, lorsqu'on coupe l'alimentation électrique d'un APLLE ou d'un ORIC, toutes les informations contenues disparaissent. Il faut donc les recopier sur un support plus stable : une DISQUETTE.

Une disquette, c'est un petit morceau de plastique, dans lequel l'information pourra être consignée magnétiquement, comme sur la bande d'un magnétophone.

L'accès est beaucoup plus rapide que sur la bande. En effet dans le magnétophone l'écriture est linéaire. Pour aller chercher une information, il est parfois nécéssaire de dérouler laborieusement toute la bande. Ce qui peut prendre beaucoup de temps.

Dans la disquette, l'information est distribuée différemment. Selon des pistes, avec un repérage selon la distance au centre et selon la position angulaire de la disquette :





Voici le lecteur de disquette.
Il transforme complêtement le travail du mixro ordinateur.

Que côute un lecteur de disquette ?

Il y a cinq ans : 5000 F.Ce prix est en train debaisser.Les lecteurs de l'ORIC ne feront guère plus de 2000 F.
Nous l'avons dit : l'Informatique proprement dite ne côute pas grand chose.Quelques centaines de francs.Ce qui limitera le baisse du prix de ces "drives",ce sera la partie mécanique.

Mais,produits en grande série,ils ne côuteront pas plus que les classiques minicassettes.

Que reste-t-il ? La "visu".Avec son écran fluorescent.
On en trouve à moins de 900 F.Et ce prix baissera.
Plus encore,il faut s'attendre à l'émergence de nouveaux produits à base de cristaux liquides.Les écrans à cristaux seront bientôt sur le marché.produits en grande série,ils ne vaudront pas plus que quelques centaines de francs.

Conclusion : sous peu,un ordinateur déja performant,avec lequel on peut apprendre énormément de choses et sur lequel on peut travailler DES ANNEES avant d'en faire le tour,pour moins de mille francs,tout compris.

Difficile d'éviter l'informatique dans ces conditions....

L'INFORMATIQUE,POURQUOI FAIRE ? (*)

L'informatique va donc envahir la planète.On trouve déja des puces un peu dans n'importe quoi.Dans les machines à laver, dans les téléphones,dans les automobiles,dans les jouets.

Mais quid de l'implantation de micro ordinateurs dans les écoles,dans les facs,entre les mains de simple quidams ?

Un ordinateur est un ENSEMBLE qui peut être comparé à un MECCANO.

Lorsque vous étiez enfant,vous avez peut-être joué avec ces jouets là.Il y avait la boite numéro un,puis la deux... Pour construire un pont transbordeur,il fallait être d'une famille riche,car ces boites ont toujours été assez chères. Le MECCANO de l'informatique est essentiellement bon marché,et ce phénomène n'ira qu'en s'accentuant.A l'heure où les prix flambent,les prix des machines baissent.Et,pour être plus précis, le rapport service offert sur prix croit à une vitesse folle.

Même les professionnels ne sont plus capables d'exploiter les capacités des machines qui apparaissent sur le marché.C'est comme si,sur simple demande,on vous livrait un nombre incalculable de boites,d'un coup.On a même plus le temps de les ouvrir,ou de lire seulement les notices,que d'autres boites apparaissent.

Considérons donc l'ordinateur comme une très belle boite de MECCANO.Que peut-on en faire ?

Un jouet à microprocesseur,une machine qui joue aux échecs, c'est l'objet "tout monté".Ca peut avoir son utilité.

Mais vous savez très bien qu'un enfant qui manipule une petit robot à distance avec une télécommande,n'en deviendra pas plus intelligent pour autant.

Ce sont les jouets "à monter soi-même" qui sont intéressants. Face à ces ordinateurs,qu'alons nous faire ?
D'abord recencer les rouages,apprendre à enfiler les roues dentées sur les axes et à placer les axes sur les supports.Apprendre à utiliser des cames,des échappements,des courroies de transmission. Apprendre le maniement des moteurs,et de tous les gadgets disponibles (et il y en a un sacré paquet dans un ordinateur !)

(*) au moment où j'écris ces lignes, un article de Libération fait état de l'effondrement de TEXAS TI dans le domaine

Un enfant qui ne commence pas par savoir comment serrer une vis et monter un écrou,ne pourra guère exploiter son jeu de MECCANO. Pour l'informatique c'est pareil.On ne peut percevoir l'intérêt du jeu que lorsqu'on a effectué un certain chemin.

LE HARD ET LE SOFT

Dans un jeu de MECCANO on trouve des éléments plus ou moins sophistiqués.Cela va de la simple barre perforée,de la vis,au moteur électrique.

Le moteur est évidemment quelquechose de plus élaboré.Il contient une mécanique,un réducteur,des bobinages.On l'utilisera en général tel quel.

Mais notre MECCANO peut aussi contenir des cellules photosensibles,ou des circuits imprimés ayant des fonctions bien définies.

Il y a donc une hiérarchie d'objets.
On peut,ceci dit,très bien envisager un jeu dont le but soit précisément de monter soi même des moteurs électriques.Auquel cas l'objet sera livré en pièces détachées : fil pour bobinage contacts en cuivre, balais,etc....

Il en est de même pour un ordinateur.

Il possède des rouages très élémentaires,qui sont les éléments constituant le microprocesseur .Nous appelerons ces objets des INSTRUCTIONS.Elles sont du type :
mettre un dans la boite X,puis comparer au contenu de la boite Y.
Si X est différent de Y alors....
Etc....
Les boites du microprocesseur s'appellent des REGISTRES.Le micro processeur est "cablé" d'une certaine manière.On peut dire qu'à ce stade (dit du langage machine),le démontage s'arrête.
Il n'est pas question d'envoyer les mains dans ces circuits de quelques microns,pour changer une connexion.

Ceci est une partie du HARD de la machine.La quincaillerie comme disznt les anglo saxons.

L'ordinateur contient également une mémoire,qui a sa propre architecture. Elle fera aussi partie du HARD.

Et le SOFT ?

C'est la manière dont on va utiliser tout cela.

Faisons une comparaison :

Dans le jeu de MECCANO il y a un moteur,une cellule photoélectrique, un petit circuit logique.Ca,on ne démonte pas.

Avec de tels éléments on peut construire des machines.Et la façon dont ces objets sont assemblés constitue le soft.

Donc,c'est clair.Vous pouvez faire tout votre soft vous même. De A à Z.Rouages par rouages.Mais vous pourrez aussi utiliser des ensembles déja assemblés.

Imaginez que dans votre jeu de Meccano vous vouliez faire intervenir un"temps mécanique",en incluant dans votre machine un réveil. Une horloge à balancier ou à ressort.

On peut monter soit-même l'horloge.Mais si cette horloge n'est pas en soi très intéressante,on peut l'acheter "toute faite" et l'inclure dans le système.

Un ordinateur est donc susceptible de recevoir du "soft". Les ordinateurs APPLE II sont multilangages.Il en est de même pour les ORIC (pour le moment ilsne savent parler que le BASIC). On peut les structurer de mille manières.Leur donner une sorte de préstructure appelée LANGAGE de programmation.

Ceci se présente sous la forme d'informations présentes dans une disquette.On fait "lire" cette disquette par l'ordinateur,et l'information est transférée dans ses intérieurs.Cela prend quelques secondes.

Au dessus de cette architecture du microprocesseur et de la mémoire ,qu'on ne peut toucher,va se superposer une autre architecture,plus élaborée,plus accessible et moins laborieuse.Et c'est à ce niveau que nous allons travailler.

On voit donc que l'ordinateur est comme les poupées russes. Au centre il existe un noyau final,indémontable.Le microprocesseur. Mais,autour,s'articulent des niveaux,en couches concentriques.

Imaginez que vous soyiez un milliardaire en promenade dans son yacht.

Vous dites à votre capitaine : amenez moi dans le port de Saint Tropez.....

Le capitaine se tourne vers son interphone (c'est un très très gros yacht) et dit:

Gouvernez au 220,machines sur un demi.

En bas le chef des machines dit à ses mécaniciens :

Mettez moi la turbine II sur 7000 tours,quatre à l'admission.

Coupez le réchauffeur.

Trois niveaux de langage.

On peut imaginer un amiral déployant toute une flotte.D'où un niveau encore supérieur.

Dans l'ordinateur,on peut se ballader à tous les étages. Intervenir directement au niveau des machines (langage machine) ou utiliser un soft tout prêt pour faire de la synthèse musicale. Au choix.....

QUEL PROFIT TIRER DU MANIEMENT D'UN ORDINATEUR ?

Alors là,c'est simple : ça dépend entièrement de vous. Quand vous connaitrez tous les rouages de la boite,et qu'on vous aura montré quelques exemples de montages,cela sera à vous de jouer. Vous pourrez choisir de recopier des montages inventés par d'autres ou vous contenter d'en utiliser d'autres,sans trop chercher à approfondir.

L'ordinateur vous servira alors à classer des objets,à les trier extrêmement rapidement,ou à mieux jouer aux échecs,ou à dessiner en perspective.

Mais le profit le plus important me semble être ailleurs. L'ordinateur vous force à réfléchir sur votre propre langage,sur votre propre pensée.

Il va vous contraindre à élaborer des démarches LOGIQUES.
Ceci dit,on peut le considérer comme une sorte de prothèse intellectuelle.
Le mot vous chqque ?
Qu'est-ce que l'écriture,sinon une MEMOIRE EXTERNE ? Qui décharge votre cerveau de tout un fatras de connaissances.Que seraient les hommes sans leurs livres ?
L'ordinateur est capable de prendre en charge des bfibes de pensée
On peut,grâce à lui,réaliser de véritables expériences,déployer un nombre fantastique de possibles.On peut aussi voit apparaitre quelquechose auquel on aurait jamais pensé.C'est une aide à la création.

En programmant un ordinateur,vous l'enrichissez.Mais en faisant l'effort de le programmer,vous VOUS enrichissez.

L'ordinateur est déja indissociable de l'enseignement des sciences.Dans quelques années les livres de physique seront diffusés,avec leur disquette,qui permettra de voir s'animer les formules.

Il est prévu d'introduire également l'ordinateur dans l'enseignement des lettres.Comme dans les enseignements artistiques.
Il y a un an ou plus,je faisais une conférence dans l'école des beaux arts d'Aix en Provence.

Un vieux prof se leva:
- n'allez pas me dire quel'ordinateur remplacera un jour l'artiste
Diable,je n'étais pas venu pour cela.Je voulais seulement avertir ces braves gens que dans peu d'années ils pourraient jouer avec des ordinateurs comme avec une palette et des tubes de couleur.
Et qu'un jour même,à travers des hologrammes de synthèse,ils pourront sculpter des objets immatériels,au laser,des formes folles,comme jamais la main de l'homme nepourra en créer,et qui ne seront faites que de lumière.

L'ordinateur ne remplacera pas non plus le musicien.Mais il y aura une autre espèce de musiciens.Qui jouera avec les formes sonores.

Ce n'est pas ma faute si les informaticiens ne sont pas des artistes,et si les "toiles" qu"ils inventent sur leurs bécanes sont des croutes.

En inventant le burin,le vernis,l'acide et le cuivre,on créait la gravure.Mais le talent n'était pas fourni avec.

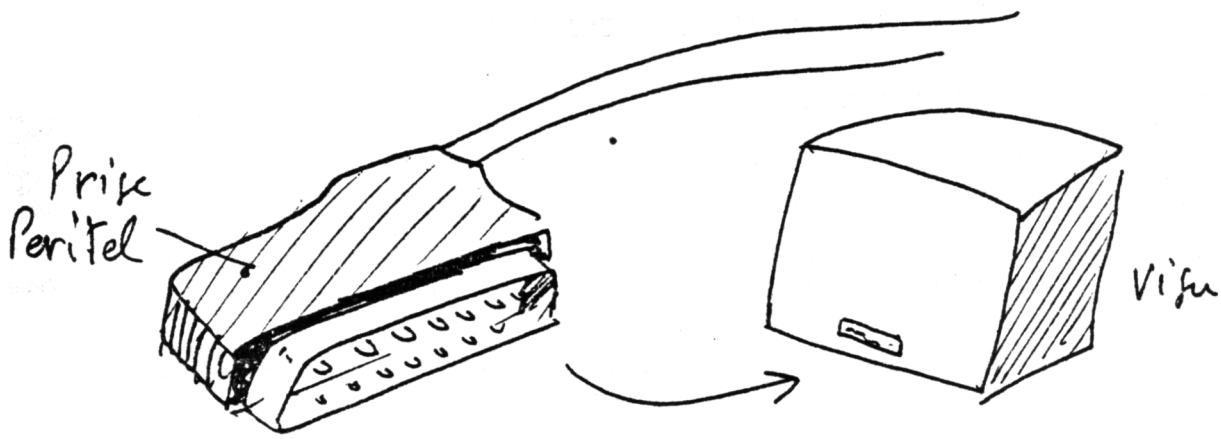
Voilà.
Au delà de tout cela,un aspect à bien méditer : si vous apprenez à vous servir de tous les rouages,parfaitement,et si vous n'avez pas un minimum d'idée,d'imagination,vous passerez à côté de la plaque.Comme avec le MECCANO.

L'informatique est exigente.Il ne suffit pas de savoir visser des boulons.Un jour,vous découvrez que,sans idées,vous tournez en rond.Alors,tâchez d'en avoir.....

Notre but est de mettre en circulation des documents pédagogiques qui puissent vous donner le plus d'autonomie possible.Il y a des gens qui ne pourront pas bénéficier d'un enseignement magistral. Il devront se dépatouiller avec ce polycopié.

Première chose : si vous le pouvez,achetez vous une machine personnelle.

L'ORIC-1 est (en 1983) ce qui offre le meilleur rapport qualité prix.Il vaut un peu plus de 2000 F.
Il est équipé d'une prise PERITEL :



Si votre téléviseur est récent ~~xxxxxxxxxxxxxxxxx~~ vous pourrez envisager de brancher directement votre ordinateur dessus.

Sinon il est possible de modifier légèrement la sortie de l'ORIC pour faire en sorte qu'il puisse attaquer directement l'antenne d'un téléviseur quelconque (noir ou couleur).L'image sera seulement un peu moins bonne.

Cette opération peut être négociée à Marseille à

ASN DIFFUSION,20 rue VITALIS. Téléphone (91) 94-15-92

Demandez monsieur TAILLEZ,et venez avec ce cours.En principe il vous fera "un prix".

Si vous n'avez pas de téléviseur avec prise PERITEL, sachez qu'il existe maintenant des téléviseurs couleurs pour 2500 F.
Enfin,en attendant de doter l'ORIC d'un lecteur de disquettes (valant lui aussi un peu plus de 2000 F) vous pourrez utiliser pour le stockage externe un simple minicassette.

Détails mercantiles,mais qui pourront aiguiller utilement les gens qui ont décidé (et qui ont les moyens) d'acheter un ordinateur.

Autre chose : Sous une forme plus attrayante et vulgarisée,le pendant BD de ce cours,chez CEDIC-NATHAN:
L'INFORMATIQUE PREMIER CONTACT et INFORMATIQUE ET LOGOTRONIQUE.(*)

Maintenant allons y.

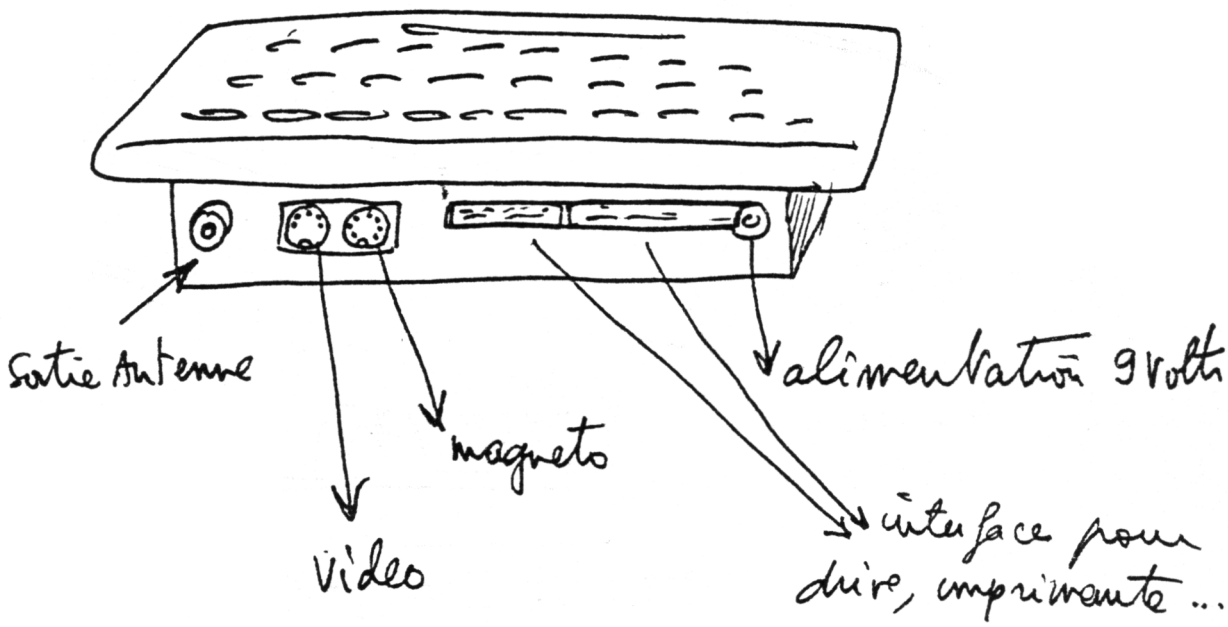
(*) Qui vont probablement être rachetés par PSI.

PREMIER CONTACT AVEC L'ORIC-1

Vous êtes peut-être dans la salle du libre service de la fac. Alors, tout est branché et vous n'aurez qu'à actionner l'interrupteur.

Si vous êtes chez vous, et que tout est en vrac, voyons ce qu'il faut faire.

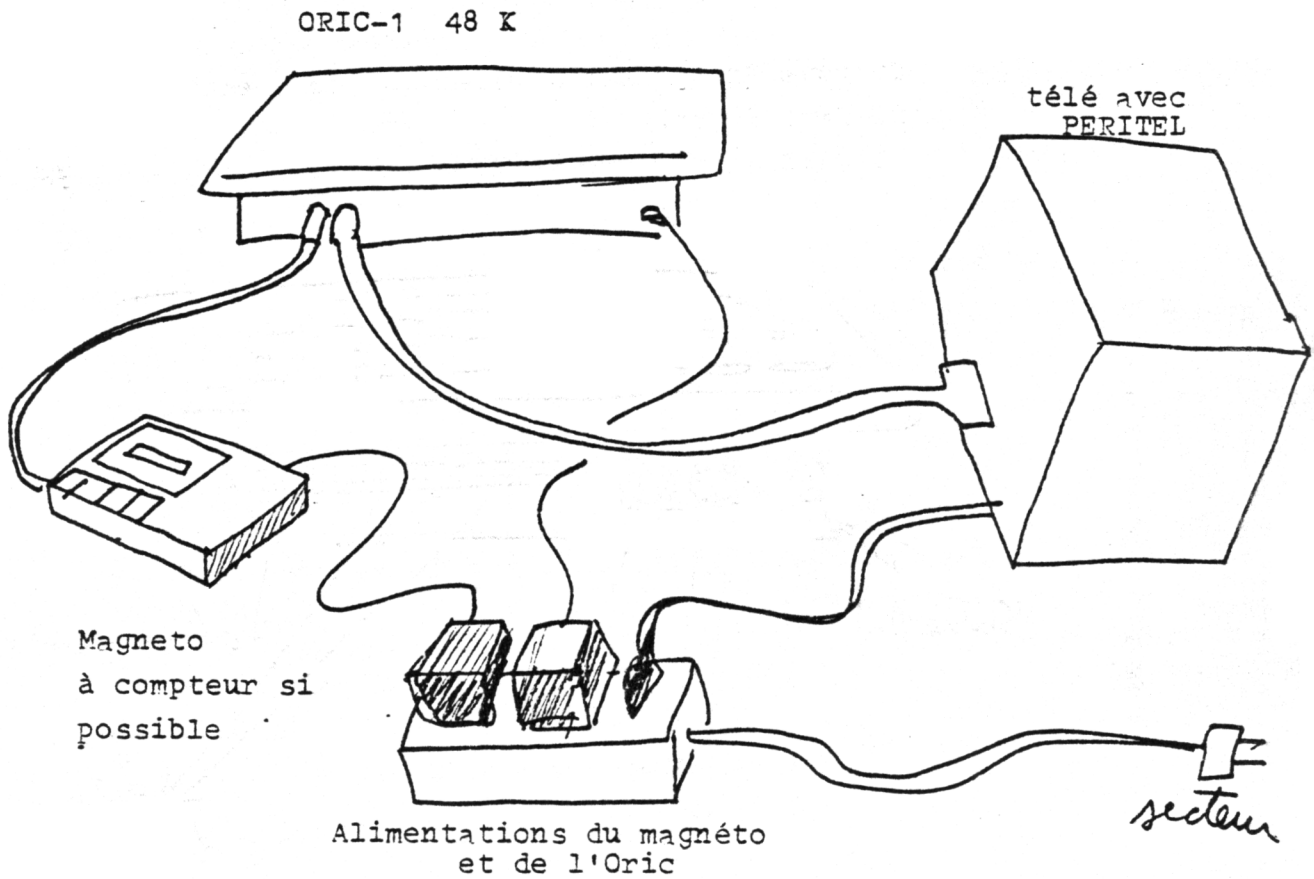
L'ORIC a un certain nombre de sortie, sur sa face arrière.



Il est livré en France avec une attaque d'antenne en système PAL (anglais). Vous ne pourrez vous en servir que si l'appareil a subi la modification, chez ASN.

A droite le branchement de l'alimentation.

Egalement visibles, les prises pour les broches PERITEL et magneto.



Voici l'ensemble branché.

Mettez sous tension.
Pendant quelques secondes vous voyez s'inscrire sur l'écran une constellation de carrés et de barres blanches horizontales. C'est normal. L'ORIC travaille. Il effectue des chargements des transferts internes. Ceci achevé, vous voyez apparaitre :

Si ça ne gaze pas, débranchez le fil alimentant l'ORIC, puis rebranchez le ça devrait aller.

ORIC EXTENDED BASIC V1.Ø
C 1983 TANGERINE

47870 BYTES FREE
Ready

Cela signifie que notre ORIC parle le "basic étendu" et que sa mémoire utilisateur laisse 47870 "bytes" libres.

ALLONS-Y:

CLS

Voilà votre première instruction. Il faut composer les lettres C L S au clavier. Le message est ainsi constitué. Mais l'ordinateur ne fait rien. Le message n'a pas encore été "envoyé".
Pour ce faire il suffit de presser sur la touche RETURN.
Et l'écran est effacé.
CLS : Clear screen (nettoyer l'écran en anglais).
Maintenant, sur votre écran :

Ready et le "curseur" clignotant.

L'absence de curseur peut être un signe de disfonctionnement.
Si cela se produit, geste simple : débranchez l'alimentation de l'ORIC (ou si vous êtes en salle éteignez l'interrupteur) .
Normalement vous devez retrouver la séquence du début.

Dans la marine, on donne d'abord les ordres, puis, on déclenche leur exécution. Exemple :

- A mollir la balancine !
- Paré...
- Envoyez !

Evidemment on ne mollit la balancine que lorsque on entende le mot "envoyez", sinon, ça peut être une catastrophe. Si la voile n'est pas bien étarquée, la baume peut aller assommer le capitaine.
SI VOUS AVEZ UN ORIC AVEC UNE TELE OU UN MONITEUR COULEUR :

PAPER INK

Ces instructions permettent de changer la couleur du "papier" et de "l'encre".

Exemple PAPER3 (et évidemment presser sur la touche return)
INK6 (id)

A partir de maintenant nous omettrons de préciser cette pression de la touche return en fin de message.

Il y a ~~sept~~ huit couleurs possibles:

0 NOIR
1 ROUGE
2 VERT
3 JAUNE
4 BLEU
5 MAGENTA
6 CYAN
7 BLANC

Je vous laisse vous amuser cinq minutes avec ce jeu des couleurs.

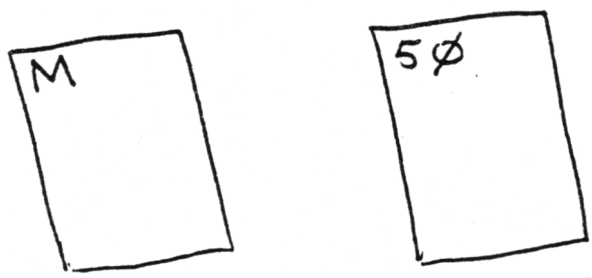
Et nous allons passer à un chapitre important.

MÉMOIRE REGISTRES

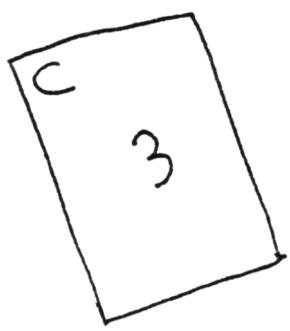
Pour nous,un ordinateur sera assimilé à un paquet de fiches rangées dans une boite.Nous appelerons ce fichier la MEMOIRE. Chaque carte sera une case mémoire,vierge.Si nous voulons consigner une information dans cette série de carte il est bon de prévoir un système de repérage.

On appelera ce système de repérage une ADRESSE.

On peut faire deux types de repérages: avec des lettres de l'alphabet ou avec des numéros :



Dans le langage BASIC ces repèrages correspondent à deux utilisations bien différentes.
Nous allons d'abord consigner des données numériques.Par exemple des nombres entiers.Et nous utiliserons un repérage à l'aide de lettres.



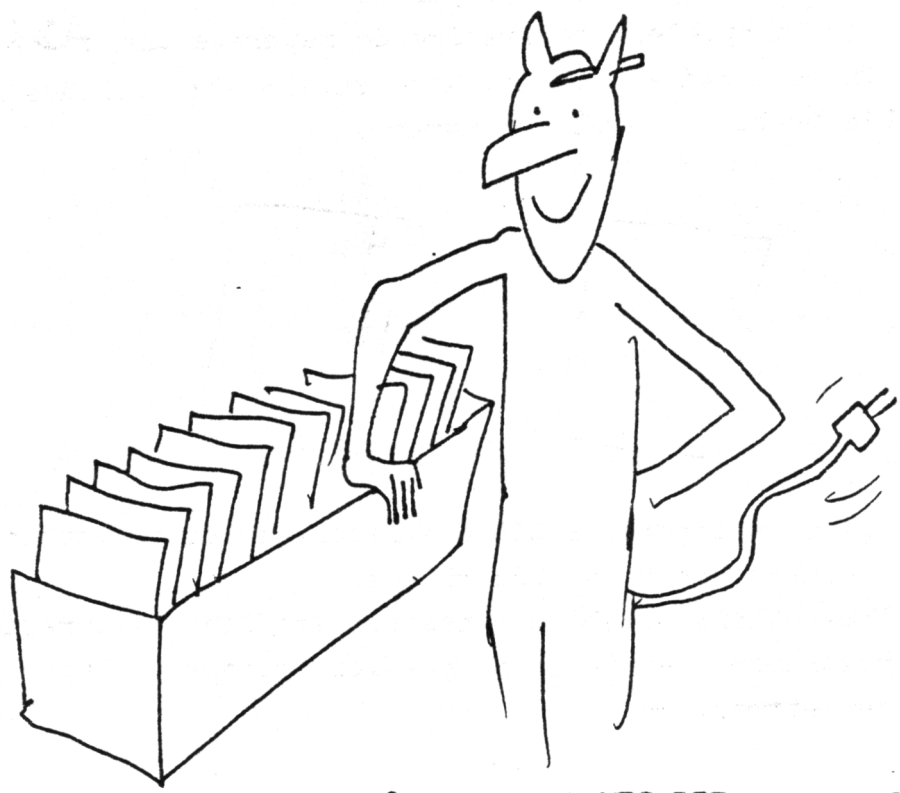
Voici par exemple une des cartes sur laquelle j'ai mis le chiffre trois,qui est la quantité numérique à stocker.Et dans le coin en haut et à gauche,j'ai mis la lettre C.

Ceci signifiant :

Mettre 3 dans C.

Ce faisant j'ai créé un REGISTRE NUMERIQUE C.

Imaginons deux personnes.L'une est l'utilisateur de la machine et l'autre l'INTERPRETEUR BASIC.C'est à dire une sorte de diable qui loge dans la machine et qui est censé comprendre les ordres que vous lui donnez :



Son nom est MEGABIT et vous le retrouverez si vous avez envie de vous détendre un peu,dans l'album L'INFORMAGIQUE,aux éditions BELIN (auteur:votre serviteur).

Eb langage BASIC (le seul que comprenne MEGABIT),pour créer unrregistre C et y mettre la valeur 3 on composera

C = 3

Bien... on peut aussi inscrire des valeurs négatives:

A=-8

On réalise ainsi des opérations D'ECRITURE EN MÉMOIRE

PRINT

?

L'opération inverse est la LECTURE..

On va interroger la mémoire pour savoir ce qui se trouve dans ces registres C et A.Et on effectuera ce questionnement à l'aide de l'ordre PRINT

Composons PRINTA
Le contenu apparait.
Même chose si on compose ?A (qu'y-a-t-il dans A ?)
Comment effacer le contenu d'un registre ? Il suffit de le remettre à zéro.En composant A=Ø
Il existe une possibilité de remettre tous les registres mémoire à zéro d'un coup

CLEAR vérifiez-le

SYNTAXE MESSAGES D'ERREUR

L'ordinateur est capable d'avaler des nombres décimaux. Composons :

Z= 5,16

Apparait :
?SYNTAX ERROR

Erreur de syntaxe.Le dialogue avec l'ordinateur est lié avec le respect d'une syntaxe parfaitement rigide.La virgule est un symbole français liée aux nombres décimaux.Or cet ordinateur est anglais....
Donc il ne comprendra que l'ordre:

Z=5.16

A chaquefois que vous ferez une bêtise,la machine réagira en affichant un message d'erreur.

MODE DE REPERAGE DES REGISTRES NUMERIQUES:

On peut utiliser deux lettres,ou une combinaison d'une lettre et d'un chiffre.Exemples :

RS = 23
G1 = - 45.6

Mais vous pouvez écrire :

RAYON = 17

Et interroger la mamoire en composant

?RAYON

Mais seules les deux premières lettres seront prises en compte. Faites l'essai en composant

?RADIS

Vous parvenez au même résultat. Il faut donc faire attention.

On cite le cas d'un importateur en matériaux de construction, qui s'étant adressé à une coopérative, avait commandé par ordinateur des tomettes, s'était vu livrer des.... tomates !

AUTRE PRECISION SUR LE ZERO.
Sur votre clavier vous avez une touche O. C'est le "O" de l'alphabet. Le zéro est ailleurs, à la suite du neuf. Il est barré. C'est le zéro des informaticiens occidentaux.

CHAINES DE CARACTÈRES

Dans la MEMOIRE de l'ordinateur, on peut stocker ce qu'on appelera des chaines de caractères. Tout ce qui figure sur l'écran est un ensemble de caratères. A chaque fois que vous pressez sur une touche, d'ailleurs, vous envoyez un ordre élémentaire qui est précisément un caractère, qui est stocké dans unemémoire d'attente. C'est seulement lorsque vous pressez sur RETURN que l'ensemble de caratères est INTERPRETE, et le message accepté ou rejeté selon que vous aurez ou non respecté la SYNTAXE.

Si Vous tapez TR=
au moment de l'interprètation du message, l'ordinateur saura que vous voulez créer un REGISTRE NUMERIQUE. MEGABIT va donc tirer une fiche, écrire dessus TR, et lire la suite du message.

Si on désire stocker des chaines de caractères on devra faire suivre le repérage du registre de donnée du signe $.

Ca ressemble au signe DOLLAR,mais cela vient de STRING,qui veut dire chaine.

Je pourrai donc composer:

N $ = "AR(87;$$+"

Et si j'interroge ma mémoire en faisant
?N $

Je verrai apparaitre AR(87;$$+

C'est à dire ma suite de caractères.

Petite précision : certains caractères sont obtenus comme les majuscules d'une machine à écrire.Par exemple ce signe $,qui est au dessus du 4.
Si on presse directement sur la touche 4 on obtient bêtement un quatre.Mais si on presse au préalable sur la touche SHIFT (qu'on doit alors tenir pressée) et qu'on refait la même opération,on verra apparaitre $ sur l'écran .

Il est évident que si on peut stocker des chaines de caractères quelconques,on peut aussi stocker :

N $ = "CERISE"

Mais l'ordinateur ne fait pas la différence entre un nom cohérent et une suite quelconque de caratères.

Il existe un programme amusant nommé ELIZA,qui tente d'analyser vos messages,en jouant le rôle d'un psychanalyste.

Si au lieu de message,vous laissez votre chat se ballader un instant sur le clavier,puis que vous pressiez sur la touche Return, l'ordinateur répondra :

JE NE SUIS PAS SUR DE TRES BIEN VOUS COMPRENDRE

Vous avez aussi remarqué que la chaine de caratère était entourée par des guillemets. C'est impératif. Des opérations comme:

```
BN $ = ALBERT
N1$="ALBERT


 G$=ALBERT"
NG="ALBERT"
```

Sont illicites et déboucheront sur un message d'erreur. Ilfaut impérativement respecter la SYNATXE § ZUT....)

LA NOTION DE PROGRAMME

Nous avons commencé, avec notre jeu de MECCANO, et nous apprenons lentement à monter quelques pièces. Cette mise en mémoire et cette lecture est aussi fastidieuse que de visser des écrous.

Un ordinateur est obligatoirement autre chose que cet ensemble de cartes-registres.

On va opérer sur ces données à l'aide d'un PROGRAMME. Mais qu'est-ce qu'un programme ?

Prenons une machine à laver. Avec un seul programme. On a une molette qu'on tourne, qui arme une minuterie. Les opérations de la séquence de lavage sont les suivantes :

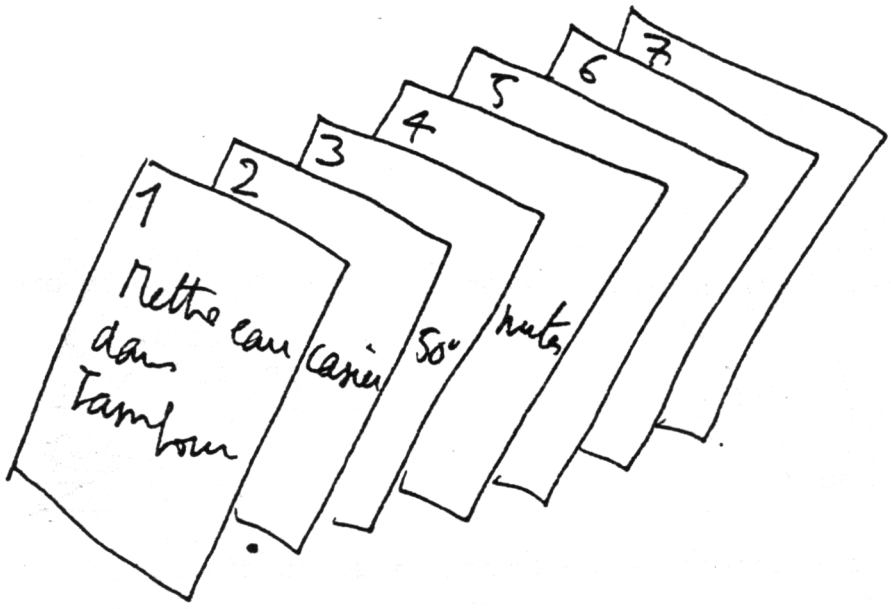
1 METTRE DE L'EAU DANS LE TAMBOUR
2 ALLER PRELEVER LA LESSIVE DANS LE CASIER
3 PORTER L'EAU A 50°
4 METTRE EN ROTATION PENDANT 10 minutes.
5 VIDER L'EAU DU TAMBOUR
6 ESSORER
7 STOP

Ces ordres élémentaires sont par exemple inscrits sur un disque.

Le dessin montre le programme en cours d'éxécution. Sur le plot 7 la machine s'arrètera d'elle même.

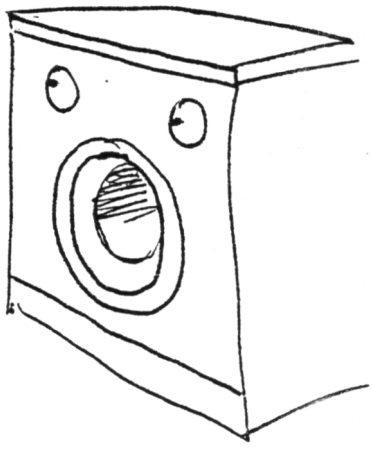
Sur ce type de machine on peut faire exécuter partiellement les programmes. Par exemple en remettant le disque surla position 6 on obtiendra seulement un essorage.

Passons à l'ordinateur. Prenons une pile de cartes dans la "boite" mémoire et écrivons non plus des lettres de repérage de registres, dans le coin haut et gauche, mais des numéros d'ordres



Bien sûr,tel quel,notre ORIC ne sait pas laver le linge.Il lui faudrait une interface ad hoc.Donc les ordres présents sur ces cartes sont des erreurs de syntaxe.

Pour que ça ne coince pas,remplaçons les par des ordres PRINT.



Dans ce modèle,cette suite de cartes constitue un modèle de programme.

Si je me mets maintenant à côté de MEGABIT et que je dis
- Allez !
Il va prendre les cartes les unes après les autres et effectuer la tâche qui est dessus.
En BASIC on déclenche l'éxécution du programme en composant

RUN

Comment "entrer" le programme dans l'ordinateur ?
Il suffit de composer les "lignes programme" :

```
1 PRINT"METTRE DE L'EAU DANS LE TAMBOUR"
2 PRINT"ALLER PRELEVER LESSIVE DANS CASIER"
3 PRINT"PORTER EAU A 50 DEGRES"
4 PRINT"METTRE EN ROTATION DIX MINUTES"
5 PRINT"VIDER EAU TAMBOUR"
6 PRINT"ESSORER"
7 END
```

Puis on fait RUN et on voit apparaitre :

```
METTRE DE L'EAU DANS TAMBOUR
ALLER PRELEVER LESSIVE DANS CASIER
PORTER EAU A CINQUANTE DEGRES
METTRE EN ROTATION DIX MINUTES
VIDER TAMBOUR.
```

Le programme a été exécuté.
Comme tout à l'heure on peut réaliser des exécutions partielles en composant par exemple RUN4,auquel cas le programme débutera à la ligne quatre.

Vius connaissez l'instruction CLS (effaçage écran).On aurait bien aimé ajouter cette instruction au programme.Hélas il n'y a plus de place...

Pour permettre cette éventuelle modification d'un programme on s'habituera à taper de dix en dix. Ceci dit, ça ne prend pas plus de place en mémoire. Car MEGABIT opère au départ sur des cartes vierges.
Au préalable nous allons effacer le programme précédent. Pour ce faire nous utiliserons l'ordre ;

NEW

qui veut dire neuf. Prêt pour un second programme.

```
10 PRINT"METTRE EAU DANS TAMBOUR"
20 PRINT"ALLER PRELEVER LESSIVE DANS CASIER"
30 PRINT" PORTER EAU A CINQUANTE DEGRES"
40 PRINT"METTRE EN ROTATION DIX MINUTES"
50 PRINT"VIDER EAU TAMBOUR"
60 PRINT"ESSORER"
70 END
```

Ah...zut, j'ai encore oublié cette instruction CLS !

Qu'à cela ne tienne, je vais l'INSERER dans le programme en composant:
5 CLS
Vous vous dites peut-être : "ça n'est pas en ordre". On va vérifier.
Nous allons "lire" le programme à l'aide de l'ordre :

LIST

Faites manuellement CLS. L'écran se vide.
Puis faites LIST, hop! le programme réapparait et l'insertion s'est opérée comme souhaité.

NB : Pour modifier une ligne erronée, retapez-la tout simplement.... pour le moment.

L'Instruction LIST peut permettre d'afficher une ligne unique.
Exemple,faites LIST 2Ø
La ligne 2Ø apparait.
Comment faire pour supprimer une ligne ?
Par exemple la ligne 3Ø.
Vous tapez simplement 3Ø et Return.C'est à dire une ligne "vide".
Et le tour est joué.

LIST -3Ø

édite le programme sur l'écran de son début à la ligne 3Ø comprise

LIST 2Ø-

Edite le programme àpârtir de la ligne indiquée

LIST 3Ø-5Ø

Edite sur l'écran le programme de la ligne 3Ø à la ligne 5Ø, toutes deux incluses.

END

L'instruction END arrète l'exécution du programme. Vous pouvez le vérifier en l'incluant, en l'insèrant n'importe où dans le programme précédent.

Ceci permet de consigner plusieurs programmes dans la mémoire. Reprenons ce parllèle avec le thème de la machine à laver. Faisons NEW pour effacer toute trace du programme précédent et écrivons :

```
200 REM LINGE SALE
210 PRINT"REMPLIR LE TAMBOUR D'EAU"
220 PRINT"ALLER CHERCHER LESSIVE DANS LE CASIER"
230 PRINT"PORTER L'EAU A SOIXANTE DEGRES"
240 PRINT"TOURNER TAMBOUR PENDANT TRENTE MINUTES"
250 PRINT"VIDER EAU"
260 PRINT"REMPLIR DE NOUVEAU TAMBOUR"
270 PRINT"TOURNER CINQ MINUTES"
270 PRINT"VIDER TAMBOUR"
280 PRINT"TOURNER TAMBOUR PENDANT CINQ MINUTES"
290 END
300 REM LINGE TRES SALE
310 PRINT" REMPLIR LE TAMBOUR D'EAU"
320 PRINT"LLER CHERCHER LESSIVE DANS CASIER"
330 PRINT"PORTER L'EAU A QUATRE VINGT DEGRE"
340 PRINT"TOURNER PENDANT TRENTE MINUTES"
350 PRINT"VIDER EAU"
360 PRINT"REMPLIR DE NOUVEAU TAMBOUR"
370 PRINT"TOURNER CINQ MINUTES"
380 PRINT"VIDER TAMBOUR"
390 PRINT"REMPLIR TAMBOUR"
400 PRINT"TOURNER CINQ MINUTES"
410 PRINT"VIDER TAMBOUR"
420 PRINT"REMPLIR TAMBOUR"
```

```
430 PRINT"TOURNER CINQ MINUTES"
440 PRINT"VIDER TAMBOUR"
450 PRINT"REMPLIR TAMBOUR"
460 PRINT"TOURNER CINQ MINUTES"
470 PRINT"VIDER TAMBOUR"
480 PRINT"TOURNER CINQ MINUTES"
490 END
500 REM LINGE DELICAT
510 PRINT"REMPLIR TAMBOUR"
520 PRINT"ALLER CHERCHER LESSIVE DANS CASIER"
530 PRINT"PORTER L'EAU A QUARANTE DEGRES"
540 PRINT"TOURNER DIX MINUTES"
550 PRINT"VIDER TAMBOUR"
560 PRINT"TOURNER CINQ MINUTES"
570 END
```

En composant RUN 200 ou RUN 300 ou RUN 500 on "appelera" trois programmes différents,corresponsant chacun à une tâche différente.

Nous allons perfectionner ce programme.

PROGRAMME AVEC ENTRÉE DE DONNÉES.

Nous allons créer un programme à options.

ECRIVONS :

```
5 CLS
10 PRINT"1- LINGE SALE"
20 PRINT"2-LINGE TRES SALE"
30 PRINT"3-LINGE DELICAT"
```

40 INPUT VOTRE CHOIX ";C

INPUT

Cette instruction arrète le cours du programme. Un questionnement est effectué.
Attention à la syntaxe ! Après l'ordre INPUT, la question posée doit fugurer entre guillemets. Puis il faut mettre un POINT VIRGULE. Et enfin indiquer le registre (ici) numérique, où la réponse doit être placée.
Quand un ordre INPUT est présent dans un programme, MEGABIT s'arrète, ouvre un tiroir, ou prend une fiche vierge sur laquelle il inscrit ici C et attend la réponse.

Placez provisoirement une ligne

45 END

Qui stoppera le programme à ce stade d'exécution. Puis faites RUN.

APPARAIT :

VOTRE CHOIX ? ■

Vous pouvez alors donner votre réponse (soit 1 soit 2 soit 3).
La ligne 45 termine l'exécution du programme.
Faites alors ? C

On voit apparaitre le contenu du registre C.
MEGABIT a donc bien fait son travail, c'est à dire créé ce registre et mis la valeur donnée dedans.

BRANCHEMENT CONDITIONNEL

Comment se servir de cette donnée C pour orienter l'exécution ?
Supprimons d'abord la ligne 45, provisoire.
Ecrivons :

```
50 IF C = 1 THEN 200
60 IF C=2 THEN 300
70 IF C=3 THEN 500
80 PRINT"ERREUR,REPONDEZ 1,2 ou 3 ! "
90 GOTO 40
```

IF...THEN

La ligne 50 signifie :

SI C est égal à 1 ALORS allez à la ligne 200

La ligne 60 : SI C = 2 ALORS ALLER EN LIGNE 300

etc...

Que se passe-t-il si la réponse donnée n'est ni 1 ni 2 ni 3. La structure de ce programme permet de lever cette ambiguité. En effet si aucun des trois tests successifs n'est satisfait les lignes 8Ø et 9Ø sont exécutées. La ligne 80 précise qu'une erreur a été commise. La ligne 9Ø représente un

BRANCHEMENT INCONDITIONNEL

GOTO

signie ALLER EN

(dans l'exemple aller en ligne 40, c'est à dire reformuler la question).

Tant que la réponse fournie ne correspondra pas à ce qui est ainsi imposé, l'ordinateur, sans se lasser, reformuler sa question. Nous avons dans ce programme, utilisé une instruction-gadget, l'instruction

REM

Dans l'exemple :
2ØØ REM LINGE SALE

L'instruction REM permet de consigner des informations repères dans les programmes. Quand une telle ligne est "effectuée", rien ne sa passe. Cela "compte pour du beurre". Cette information n'apparait que lorsqu'on utilise l'instruction LIST, lorsqu'on fait un LISTING. Mais ce gadget est bien commode pour pouvoir s'y retrouver.

Ce programme de lavage de linge n'est pas spécialement excitant. Mais j'ai choisi cet exemple pour quel'on garde bien en tête que l'ordinateur effectue des TACHES.

SOUS PROGRAMME

L'informatique implique un souci constant d'ECONOMIE.
Dans le programme de la page 30 on constate que des opérations reviennent plusieurs fois: REMPLIR LE TAMBOUR D'EAU,TOURNER CINQ MINUTES,etc...

Un système de SOUS-PROGRAMME permet d'éviter cette fastidieuse réécriture.

Considérons la tâche:

```
PRINT"TOURNER CINQ MINUTES"
```

Qui revient de nombreuses fois.

Ecrivons :

```
1ØØØ REM ROTATION DU TAMBOUR
1Ø1Ø PRINT"TOURNER CINQ MINUTES"
1999 RETURN
```

Dans le programme,à chaque fois que nous rencontrerons la tâche TOURNER CINQ MINUTES nous la remplacerons par

GOSUB

```
GOSUB 1ØØØ
```

Concrètement,on écrira:

```
27Ø GOSUB 1ØØØ
4ØØ GOSUB 1ØØØ
43Ø GOSUB 1ØØØ
46Ø GOSUB 1ØØØ
48Ø GOSUB 1ØØØ
56Ø BOSUB 1ØØØ
```

A chaque fois que cet ordre GOSUB sera rencontré,il y aura APPEL DU SOUS PROGRAMME CORRESPONDANT.Le cours du programme sera suspendu et MEGABIT,toutes affaires cessantes,s'en ira effectuer la sous-tâche décrite dans le sous programme.

ATTENTION ! UN SOUS-PROGRAMME PEUT EN APPELER UN AUTRE...

Le sous programme est l'âme de l'informatique.En fait TOUT est sous programme.Le travail du programmeur consiste à dégager sans cesse son PROGRAMME PRINCIPAL en renvoyant des tâches qui apparaissen annexes dans des sous programmes.

Effaçons notre programme à l'aide de NEW et reprenons tout cela à zéro.

Ces différents programmes de lavage,finalement,ce sont des sous programmes.

Nous allons réécrire notre programme MENU

```
5 CLS
10 PRINT"LAVAGE DE VOTRE LINGE"
20 PRINT"1-SALE"
30 PRINT"2-TRES SALE"
40 PRINT"3-DELICAT"
50 INPUT"VOTRE CHOIX ";C
```

ON.... GOSUB

```
60 ON C GOSUB 1000,2000,3000
70 END
```

La ligne soixante signifie
SI C=1 effectuer le sous programme 1000
Si C=2 ffectuer le sous programme 2000
si C=3 effectuer le sous programme 3000
On peut en mettre autant qu'on en veut,ou presque...
Ce faisant,on ne traite pas le cas où l'utilisateur aurait la mauvaise idée de répondre par autre chose que 1,2,3....
Mais ne compliquons pas.

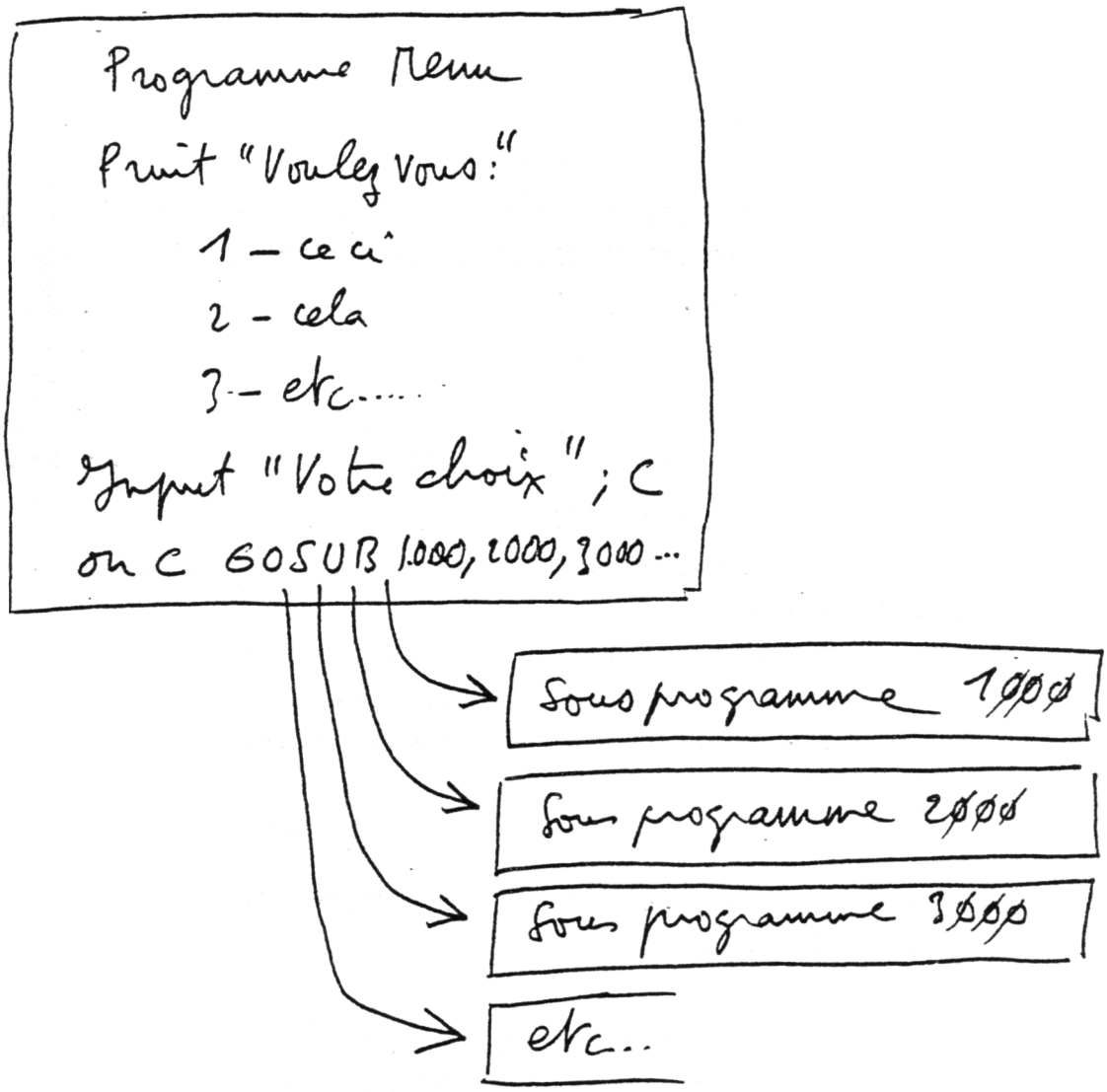
A ce stade on écrira :

```
1000 REM LINGE SALE
1999 RETURN
2000 REM LINGE TRES SALE
2999 RETURN
3000 REM LINGE DELICAT
3999 RETURN
```

La structure de notre programme est en place.Pour le moment ces sous programmes sont vides.Mais ça n'a pas d'importance;nous les remplirons plus tard.

C'est une façon de travailler,une habitude à prendre.
Ces strutures avec programme MENU et appel de sous programmes sont très classiques.Et très commodes.

A ce stade,si vous lancez le programme avec un RUN,apparemment rien ne se passera.Mais si vous n'avezpas de message d'erreur, c'est que la structure d'exécution a été correctement écrite.



Ceci fait on peut s'occupper des sous programmes.

Le sous programme 1000 décrit le lavage du linge sale. Dans cette séquence nous avons une phase de rinçage. Dans laquelle on vide le tambour, on le remplit de nouveau, on tourne pendant cinq minutes et on vide à nouveau.

Nous renconterons cette séquence dans les trois sous programmes A noter que dans le sous programme "LINGE TRES SALE", nous aurons quatre rinçages successifs.

Ceci suggère de créér un sous-sous-programme "rinçage".

```
10000 REM RINCAGE
10010 PRINT"VIDER TAMBOUR"
10020 PRINT"REMPLIR TAMBOUR"
10030 PRINT"TOURNER PENDANT CINQ MINUTES"
10040 PRINT"VIDER TAMBOUR"
10999 RETURN
```

Pourquoi en 10000 ?
Nous avons de la place. Nous pouvons donner des adresses de lignes programmes de 1 à 40000.
Il est bon de situer les sous programmes dans des lignes 1000, 2000, 3000 etc....
Et quand on a des niveaux de sous programmes supérieurs, de les situer en 10000, 11000, 12000, etc....
NB vous pourriez à partir de ces sous-sous-programmes, appeler d'autres sous-sous-sous-programmes, etc...
Mais il est rare qu'on dépasse ce stade.

On va donc écrire les sous programme 1000 comme suit :

```
1000 REM LINGE SALE
1010  PRINT"REMPLIR LE TAMBOUR D'EAU"
1020 PRINT"ALLER CHERCHER LESSIVE DANS CASIER"
1030 PRINT"PORTER L'EAU A SOIXANTE DEGRES"
1040 PRINT"TOURNER PENDANT TRENTE MINUTES"
1050 GOSUB 10000
1060 PRINT"TOURNER PENDANT CINQ MINUTES"
1999RETURN
```

Passons au programme "LINGE TRES SALE"

On écrira :

```
3000 REM LINGE TRES SALE
3010 PRINT"REMPLIR LE TAMBOUR D'EAU"
3020 PRINT"ALLER CHERCHER LESSIVE DANS CASIER"
3030 PRINT"PORTER L'EAU A QUATRE VINGT DEGRES"
3040 PRINT"TOURNER PENDANT TRENTE MINUTES"
```

```
3050  GOSUB 10000 : GOSUB 10000 : GOSUB 10000 : GOSUB 10000
3060 PRINT"TOURNER CINQ MINUTES"
3999 RETURN
```

A la ligne 3050 on voit figurer les quatre essorages.
Que signifie cette écriture ?
On peut consigner plusieurs lignes sur une seule,en séparant les opérations par deux points.Ici ces ordres sont des appels de sous Programmes mais on aurait pu mettre n'importe quel ordre correspondant au langage BASIC.

Ca a déja meilleure allure.

Mais on pourrait procéder autrement.

BOUCLES FOR...NEXT

Essayons de standardiser encore les opérations.Dans ces tâches de lavage on commence par emplir le tambour d'eau.Puis on va chercher la lessive dans le casier.Puis on chauffe à une température T. Enfin on tourne pendant un temps TR.On fait N rinçages,on essore,fin. Il y a donc trois paramètres de lavage.

La température de lavage T
Le temps de lavage TR
;Le nombre d'essorages N

Imaginons que ces trois renseignements soient consignés dans des registres T , TR , N .

Ces registres deviennent alors des VARIABLES.
Parce que leur contenu peut changer d'un lavage à l'autre.

Le programme standard de lavage serait ainsi :

Remplir le tambour d'eau
Porter à une température T
Tourner tembour pendant TR minutes
Faire N rinçages
Vider
Fin.

Reprenons notre programme et effaçons tout sauf le menu,c'est à dire les lignes 5 à 7Ø.
Ceci dit,en écrivant ce manuel,nous nous sommes trouvés devant un véritable casse-tête.

Sur l'APPLE II,par exemple,il existe une instruction très commode : DEL,qui permet d'effacer des paquets de lignes.On l'utilise comme l'instruction LIST.Sur l'APPLE quand on écrit :
DEL 1ØØØ,2ØØØØ toutes les lignes se situant entre 1ØØØ et 2ØØØØ sont effacées d'un coup.
DEL 1ØØØ, efface tout à partir de la ligne 1ØØØ comprise.
Sur ce .... d'ORIC-1 pas moyen de trouver une instruction équivalente
Il existe bien une touche DEL sur le clavier,mais elle a un autre usage.J'ai bien essayé DEL 1ØØØ- et DEL 1ØØØ-1ØØØØ : SYNTAX ERROR !
Le temps presse.Rien dans ce fichu manuel de.... !
Deux hypothèses : ou ça n'existe pas (ce qui serait une sacrée lacune),ou le traducteur français a oublié de mentionner cet outil important.

Espèrons que nous arriverons à élucider cette question agaçante.
En attendant,pour restructurer ce programme,mieux vaut faire NEW et réécrire ces lignes 5 à 7Ø.Quand même....

Ceci fait nous allons réécrire le programme différemment.

Voici les nouveaux sous-programmes :

```
1000 REM LINGE SALE
1010 T=60
1020 TR=30
1030 N=1
1040 GOSUB 10000
1999 return

2000 REM LINGE TRES SALE
2010 T = 80
2020 TR = 30
2030 N = 4
2040 GOSUB 10000
2999 RETURN

3000 REM LINGE DELICAT
3010 T = 40
3020 TR=10
3030 N=1
3040 GOSUB 10000
3999 RETURN
```

Le programme lavage !

```
10000  REM LAVAGE
10010 PRINT"REMPLIR TAMBOUR d'EAU"
10020 PRINT"ALLER CHERCHER LESSIVE DANS CASIER"
10030 PRINT"PORTER L'EAU A ";T;" DEGRES"
10040 PRINT"tourner ";TR;" MINUTES"
10050 FOR I=1 TO N
10060 PRINT"VIDER TAMBOUR"
10070 PRINT"REMPLIR TAMBOUR"
10080 PRINT"TOURNER CINQ MINUTES"
10090  NEXT I
10100 PRINT" VIDER LE TAMBOUR"
10110 PRINT"TOURNER CINQ MINUTES":REM ESSORAGE
10999 RETURN
```

boucle d'essorage

L'écriture devient beaucoup plus compacte. Tout tient sur une page.
L'ordre PRINT permet d'afficher plusieurs choses bout à bout.
A condition de séparer ces choses par un point virgule.

N'oubliez pas entre les guillemets,dans les lignes 10030 et 10040 de mettre des blancs pour éviter que les impressions successives ne soient collées les unes aux autres.

La ligne 10050 signifie :

Pour I allant de 1 à N

La ligne 10090 signifie "passer à la valeur suivante de I."

Remarque : dans les options 1 et 3 le nombre de rinçages est 1.

La boucle FOR..... NEXT admet très bien cette limite.Et la têche est effectuée pour I variant de 1 à.... 1,c'est à dire une seule fois.Ca passe....

Quelques conseils.

Le point faible de l'ORIC c'est son clavier.Il pourra vous arriver de coincer une touche.Auquel cas vous verrez un caractère se répèter indéfiniment sur l'écran.Ne paniquez pas.Vous ne risque: absolument rien.Avec sang froid,décoincez la touche.

Vous pouvez aussi vous tromper en tapant un caratère.Pour reven: en arrière utilisez la touche DEL à droite du clavier.Il y a retour d'une case du curseur,avec annulation du caractère frappé.

En pressant plusieurs fois consécutives sur cette touche,vous pouvez effaçer une série de caractères erronés.

Sur l'APPLE II ont peut tout taper sans mettre de "blancs".

L'interprèteur est assez malin pour s'y retrouver.Celui de l'ORIC est de ce côté là un peu moins perfectionné.N'hésitez donc pas à en mettre plutôt plus que moins.Cela vous évitera des déboires à l'éxécution.

Execution du programme.Vous faites RUN.Vous choisissez une des options et la liste des opérations simulées est affichée.

EXERCICES SUR CETTE PREMIERE PARTIE DE COURS

BUREAUTIQUE :
Vous allez écrire un programme qui écrit automatiquement une lettre sur l'écran,portant le texte suivant :
MONSIEUR,
J'AI BIEN RECU VOTRE HONOREE DU ONZE COURANT
ET JE VOUS EN REMERCIE.
QUANT A NOTRE ARTICLE 9013-A,NOUS NE L'AVONS
PLUS EN STOCK,HELAS.

NOUS POURRONS PALLIER CETTE RUPTURE DU STOCK
EN DATE DU 2/05/83.

AVEC NOS EXCUSES

R.DUCLAMPIN.

Le programme n'est qu'une suite d'ordre PRINT.Lorsque que vous faites RUN,ce texte doit s'afficher sur l'écran.

Deuxième étape.Programme avec entrée de donnée.
Faites NEW et composez un nouveau programme.La première partie comporte des étapes d'acquisition de données,à travers des instructions INPUT.Inutile de vous embarquer dans des sous programmes pour un truc aussi simple.

L'ordinateur commencera par poser des questions comme :
"DATE DE LA LETTRE DU CLIENT"

"REFERENCE ARTICLE DEMANDE"

"DATE POSSIBLE LIVRAISON"
Dans cette "saisie" l'input sera suivi du questionnement,entre guillement,puis du point et virgule.Les renseignements seront saisis comme des chaines de caratères.

Exemple :
INPUT"DATE DE LA LETTRE DU CLIENT ";D$
Etc...
Ces renseignements entrés dans la machine la suite du programme écrira la même lettre,mais en exploitant les renseignements mis en mémoire.

Exemple :

PRINT"J'AI BIEN RECU VOTRE HONOREE DU ";D$

Etc....

EXERCICE SUR LES BOUCLES FOR NEXT

Je précise : il est toujours inutile ici de mettre des sous programmes. Ne vous embrouillez pas l'esprit inutilement. Ces exercices sont progressifs.

Vous allez faire un programme pour vous endormir. Pour "compter les moutons".

Voici le résultat escompté de ce programme:

COMBIEN
~~COMIEN~~ DE MOUTONS ?

D'abord un questionnement. Avec un INPUT et une création d'un registre nombre de moutons NM.

Puis une boucle for next qui affiche les moutons.

Précision : le premier mouton pose un problème d'orthographe.

En effet il faut que sur écran apparaisse :

1 MOUTON

2 MOUTONS

3 MOUTONS

etc....

C'est simple : commencez par afficher "1 MOUTON" puis le reste en ~~exxxi~~ écrivant FOR I = 2 TO NM

Ce qui fait commencer l'écriture au deuxième mouton.

Vous pouvez vous amuser à inclure dans votre boucle des signaux sonores correspondant à PING ZAP SHOOT .

Variante.

A la fin de l'affichage de ces NM moutons on peut mettre le questionnement suivant.

EST-CE QUE VOUS DORMEZ ?

La réponse étant consignée dans un registre R $ (attention, c'est une chaine de caractères).

La seule réponse Qui entraine un retour vers le debut de programme est la réponse NON.
On mettra donc un test :
IF R $ ="NON" THEN suivi de l'adresse correspondant au début du programme.Par exemple ce CLS qui efface l'écran,puis est suivi du questionnement.

EXERCICE SUR LES SOUS-PROGRAMMES

Vous allez construire un programme de synthèse sonore.
Il y aura un menu,puis des sous programmes de production de sons.

Le menu affichera :

VOULEZ-VOUS PRODUIRE

1- DES PING
2-DES ZAP
3-DES SHOOT

VOTRE CHOIX ?

On répond 1,2 ou 3.

Puis une question :

COMBIEN ?

La réponse étant le nombre de PING de ZAP ou de SHOOT .
Ces deux données étant entrées,le programme devra à partir de boucles FOR NEXT qui iront de 1 à N engendrer les nombre de bruits voulu.
Appelons N le nombre de bruits.
Les bruits sont engendrés par précisément ces instructions PING,ZAP,SHOOT.

Indication : si vous voulez bien séparer les bruits successifs,

mettez après chacun un ou plusieurs PRINT
Exemple :
dans votre boucle

```
140  PING
150 PRINT:PRINT:PRINT:PRINT
160 NEXT I
```

Suivant le nombre de PRINT,vous pourrez règler le rythme de cette séquence sonore.

CALCUL D'IMPÔT.

Voici un premier exemple d'application pas trop sotte de l'informatique AT HOME.
Lorsque l'impôt sur le revenu est calculé,on se sert du revenu imposable R
Chaque famille a un nombre de parts NP
Une part pour un célibataire
Une part et demie pour un célibataire ayant un enfant à charge
(les enfants à charge représentent une demie part)
Deux parts pour un couple marié
Deux parts et demie pour un couple avec un enfant
Trois parts pour un couple avec deux enfants,etc....

Le quotient familial QF est égal au revenu imposable,divisé par le nombre de parts.L'impôt se calcule ainsi "par tranches".Il est progressif en fonction de ce quotient QF=R/NP
Voici un mode de calcul correspondant à une vieille feuille d'impôt

Les chiffres sont à réviser.
Questionnement : quel est votre revenu imposable R ?
Questionnement : etes vous marié ?
Questionnement : avez vous des enfants à charge ? Combien ?

NA est le nombre d'adultes (1 pour un célibataire,2 pour un couple)
NE est le nombre d'enfants à charge.
Le nombre de parts NP est égal à NA + NE/2

(Ah,en informatique la division est représentée non par par les deux points,mais par le "slash" /).

ALGÈBRE ÉLEMENTAIRE

+ Addition
− Soustraction
/ DIVISION
= signe égal
( ) parenthèses
* multiplication

Le quotient familial est alors égal à R/NP

Le barème (1982) de calcul de l'impôt est le suivant :

SI QF <= 9890 F I = 0
Si QF >9890 F et <= 10340 F alors I=(R x 0,05) − (494,5 x NP)
Si QF>10340 et QF<=12270 F alors tranche à 10%. I=(R x 0,1) − (1011,5 x NP)
Si QF>12270 et QF<=19410F alors tranche à 15%. I=(R x 0,15) − (1626 x NP)
si QF>19410F et QF<=24950F alors tranche à 20%. I=(R x 0,2) − (2595,5 x NP)

si QF > 24950F et QF <= 31360F alors
tranche 25% $I = (R \times 0,25) - (3843 \times NP)$

si QF > 31360F et QF <= 37940F alors
tranche 30% $I = (R \times 0,3) - (5411 \times NP)$

si QF > 37940F et QF <= 43770F alors
tranche 35% $I = (R \times 0,35) - (7308 \times NP)$

si QF > 43770F et QF <= 72940F alors
tranche 40% $I = (R \times 0,4) - (9496,5 \times NP)$

si QF > 72940F et QF <= 100320F alors
tranche 45% $I = (R \times 0,45) - (13143 \times NP)$

si QF > 100320F et QF <= 118660F alors
tranche 50% $I = (R \times 0,5) - (18159,5 \times NP)$

si QF > 118660F et QF <= 135000F alors
tranche 55% $I = (R \times 0,55) - (24092,5 \times N)$

si QF > 135000F alors
tranche 60% $I = (R \times 0,6) - (30842,2 \times N)$

Ces chiffres sont évidemment à revoir puisqu'il existe maintenant je crois une tranche à 70 %. Et un impôt sur les grandes fortunes et un impôts sur l'impôts....
Bon, restons en là.

Les lignes se lisent :

Si Q >9890F or Q <= 10340F

par exemple :
SI QF EST SUPERIEUR A 9890 F ET INFERIEUR OU EGAL A 10340 F....

< INFERIEUR A

<= INFERIEUR OU EGAL A

> SUPERIEUR A

>= SUPERIEUR OU EGAL A

<> DIFFERENT DE

Un test peut porter sur plusieurs conditions simultanées.

ALGÈBRE BOOLÉENNE

AND OR

En utilisant les éléments AND ("et") et OR ("ou")
On peut créer des tests disant :

SI CECI ET CELA ALORS
SI CECI OU CELA ALORS.....

Nous allons utiliser ces nouveaux éléments,des nouveaux rouages de programmation,pour construire ce programme.

Première partie : acquisition des informations nécéssaires au calcul de l'impôts.

```
10 CLS : REM EFFACAGE ECRAN
20 INPUT "REVENU IMPOSABLE ";R
30 PRINT "ETES-VOUS:"
40"1-CELIBATAIRE"
50"2-MARIE"
60 INPUT"VOTRE SITUATION DE FAMILLE:";NA
70 IF NA<>1 AND NA<>2 THEN PRINT"REPONSE ERRONEE":GOTO 30
```

La ligne 70 est une des multiples façons d'éviter les réponses non conformes,style 1,5 (personnes à moitié mariées).
Elle signifie:
SI NA EST DIFFERENT DE 1 ET NA DIFFERENT DE 2 ALORS .....

```
80 PRINT"NOMBRE D'ENFANTS A CHARGE"
90 PRINT"SI VOUS N'EN AVEZ PAS,REPONDEZ 0"
100 INPUT NE
```

(pas obligatoire de mettre le questionnement après l'INPUT)

```
110 REM CALCUL DE  NOMBRE DE  PARTS NP
120 NP = ( NA + NE/2  )
130 PRINT"NOMBRE DE PARTS = "; NP .
140 PRINT
150 REM CALCUL DU QUOTIENT FAMILIAL QF
160 QF = R/NP
170 PRINT"QUOTIENT FAMILIAL = ";QF
180 PRINT
```

Je mets des prints pour créer des lignes vides. Pour moins serrer les textes affichés.

```
190 REM CALCUL DE LiIMPOT SUR LE REVENU
200 IF QF < 9890          THEN T=0:GOTO 1000
210 IF QF  >9890 AND QF <= 10340 THEN T=5:I=R*0.05 - 494.5 * NP:
GOTO 1000
220 IF QF >10340 AND QF <=12270 THEN T=10 :
I=R*0.1 - 1011.5*NP : GOTO 1000
230 IF QF >12270 AND QF <= 19410 THEN T=15:
I = R * 0.15 - 1625*NP : GOTO 1000
230 IF QF >19410 AND QF <= 24950 THEN
T = 20 : I = R * 0.2 - 2595.5 * NP : GOTO 1000
240 IF QF > 24950 AND QF <= 31360 THEN
T=25 : I = R * 0.25 - 3843 * NP : GOTO 1000
```

```
250 IF QF > 31360 AND QF <= 37940 THEN
T=30 : I = R * 0.3 -5411 * NP : GOTO 1000
260 IF QF > 37940 AND QF <= 43770 THEN
T= 35 : I = R * 0.35 - 7308 * NP : GOTO 1000
270 IF QF > 43770 AND QF <= 72940 THEN
T=40 : I=R * 0.4 - 9496.5 * NP : GOTO 1000
280 IF QF > 72940 AND QF <= 100320
THEN T =45 : I = R * 0.45 - 13143.5 * NP : GOTO 1000
290 IF QF > 100320 AND QF <= 118660 THEN
T=50 : I = R * 0.5 - 18159.5 * NP : GOTO 1000
300 IF QF > 118660 AND QF <= 135000 THEN
T = 55 : I = R * 0.55 - 24092.5 * NP : GOTO 1000
310 IF QF > 135000 THEN T= 60 : I = R * 0.6
- 30482.5 * NP
1000 PRINT "VOTRE TAUX D'IMPOSITION ";
T; " POUR CENT"
1010 PRINT
1020 PRINT "IMPOT SUR LE REVENU"; I; " F"
1030 END
```

Le calcul est une succession de tests.

Dans chaque ligne de test on vérifie si vous êtes dans cette "tranche".

Si oui,tout ce qui suit est exécuté,les opérations étant séparées par deux points.
Définition du taux T
Calcul de l'impôt I

Si on a détecté la tranche,inutile de faire les tests suivants.

On a alors un BRANCHEMENT GOTO qui amène en fin de programme avec affichage du résultat.

SAUVER UN PROGRAMME

De deux choses l'une : ou vous disposez d'un minicassette,ou votre ORIC est connecté à un lecteur de disquette.
Premier cas,vous positionnez votre bande sur une portion vierge.
Vous notez la cote de cette position à l'aide du compteur (préalablement remis à zéro après réenroulement de la bande à l'aide de la touche ad hoc !)
Mettez la tonalité sur AIGUS.Et la puissance au maxi.
Utilisez la touche PAUSE pour bloquer le déroulement de la bande.

Puis enfoncez la touche ENREGISTREMENT.
Au clavier de l'ORIC,composez CSAVE"IMPOT"
Relachez la touche PAUSE en déclenchant le déroulement de la bande,et aussitôt après pressez sur Return.Le programme est alors stocké.
Quand l'ORIC "vous rend la main" arrêtez le défilement de la bande.

Pour charger de nouveau ce programme,commencez par positionner la bande au bon endroit,avec un poil de marge (le repérage n'est pas ultra précis).
Enclenchez la PAUSE,puis pressez sur la touche mettant en marche le magnétocassette.
Au clavier de l'ORIC,composez:

CLOAD"IMPOT"

Relachez la touche pause et immédiatement après pressez sur Return.
L'ORIC affiche SEARCHING.....
Ce qui veut dire JE CHERCHE.

Si le programme est correctement chargé,l'ORIC rend la main.
Et c'est prêt à tourner.Arrètez alors le défilement de la bande.

Si vous avez oublié le nom du programme,vous pouvez charger ce qui se présente sur la bande en faisant simplement CLOAD""
L'ORIC chargera le premier programme rencontré.

Si vous avez un lecteur de cassette,l'ordre de SAUVEGARDE sera vraisemblablement

SAVE IMPOT

et l'ordre inverse de chargement

LOAD IMPOT

Mais comme nous n'avons pas encore reçu ces lecteurs de disquettes nous ne pouvons que reproduire à titre indicatif les ordres de l'APPLE II.Voir éventuellement manuel.

INCONVENIENTS DE CES CASSETTES.

C'est LENT.C'est LINEAIRE.Si vous avez perdu un programme vous pouvez être contraint d'explorer laborieusement toutes vos cassettes avec des CLOAD"" jusqu'à ce que vous tombiez sur la bonne bande et sur le bon endroit.Ca peut vous prendre...la journée !

Les lecteurs de disquettes réalisent une exploration de cette mémoire externe infiniment plus rapide. Et parx ailleurs un ordre permet d'en connaitre le CONTENU.

Sur l'APPLE II cet ordre est CATALOG.

On ne connait pas encore les ordres types des lecteurs ORIC. Mais il y aura sûrement quelquechose du même genre.

L'ensemble des ordres spécialement liés à l'emploi des disquettes constitue ce qu'on appelle le D.O.S. ou

DISK OPERATING SYSTEM

Ce DOS accroit considérablement les possibilités d'un ordinateur.

Nous verrons plus loin comment simuler cet ordre CATALOG avec une cassette, en créant un programme donnant le contenu.

Ceci dit, nous n'insisterons pas sur cet emploi de cassettes, car ce n'est qu'un pis aller. Logiquement, les cassettes vont disparait dansles années à venir, pour être totalement remplacées par les disquettes. Le prix deviendra abordable quand la concurrence se mettra à jouer. Pour le moment (1983) le prix d'un lecteur de disquettes APPLE II est artificiellement maintenu aux alentours de 4000 F. Il s'effondrera dès la sortie d'un matériel concurrent.

NOTION D'ALGORITHME

Jusqu'ici nous n'avons vu que des applications assez rustiques de l'ordinateur. La notion de tâche est bien apparue, ainsi qu'un mode de travail avec des options.

Les entrées de données, en cours d'éxecution, nous ont permis d'illustrer un mode CONVERSATIONNEL.

On peut évidemment utiliser un ordinateur pour calculer son impôt sur le revenu. Mais on peut aussi faire ce calculà la main en quelques minutes. A condition d'avoir les formules.

On aimerait déboucher sur un emploi de l'ordinateur qui rend un service plus spécifique, qui aide à résoudre un problème difficile à traiter " à la main".

La façon de résoudre un problème s'appelle un ALGORITHME. Dans l'argot des taupins (candidats aux grandes écoles) ceci s'appele aussi une DEMERDANTE.

Nous avons retenu un exemple qui, à dessein, est lié à la vie de tous les jours. L'EMPRUNT.

Quand un simple quidam veut emprunter de l'argent, il lui faut se rendre dans une banque. Il fixe alors le montant de la somme qu'il souhaite emprunter. Et le nombre d'années au long desquelles les remboursements seront échelonnés. Le banquier feuillète alors des tables et lui fixera le montant du remboursement mensuel.

Si le contrat est passé,le client aura sans doute bien des surprises s'il désire effectuer un remboursement anticipé.Au bout de quelques années seulement.

- Mais,je dois rembourser presque toute la somme ?
- Eh oui,pendant ces premières années,vous n'avez payé que des intérêt.Ce n'est qu'en fin d'emprunt que la dette se trouve épongée.

Le quidam reste perplexe.La chose est connue.le travail ci après mettra bien en évidence pourquoi il en est ainsi.

Un des paramètres "sensibles" de l'opération emprunt est le taux d'escompte.C'est à dire le taux d'intérêt.

Les banques ont pas mal monté ce taux ces dernières années, pour freiner la consommation intérieure en France.Comme on le verra l'accroissement de ce taux est très dissuasif.

REAGAN a monté le taux d'emprunt de ses précieux dollars. Ce qui a entrainé une montée du cours de sa monnaie et aussi provoqué une réaction en chaine dans les taux pratiqués par les pays de la "zône dollar".

COMMENT MARCHE L'EMPRUNT ?

Soit S la somme d'argent empruntée.

Soit alpha le taux d'escompte,le taux d'intérêt,mensuel.

Soit RM le remboursement mensuel.

Chaque mois la dette S est passée à

$$S(1+\alpha) - RM$$

Constatation évidente : si $RM = \alpha S$ la dette reste constante. Le temps de remboursement est infini.

Pour que la dette soit épongée un jour il faut que

$$R > \alpha S$$

Le mois suivant la dette passera à :

$$[S(1+\alpha)-RM](1+\alpha)-RM$$

et, le mois suivant à :

$$\{[S(1+\alpha)-RM](1+\alpha)-RM\}(1+\alpha)-RM$$

etc....

La formule algèbrique donnant le montant de la dette au bout de N mois, si N est grand, devient interminable.

Ce qu'on cherche, c'est la valeur de RM correspondant à un nombre NM de mois donné.

La résolution algèbrique d'un telsystème a quelquechose de désespéré.

Or c'est précisément le renseignement que veut le client. "Si je veux emprunter sur NM mois, combien aurai-je à payer chaque mois ?"

Le banquier a sa table avec lui. Il est à même de fournir le résultat. Le calcul a été fait d'avance. Pour tous les taux d'intérêt possibles.

Imaginons que nous soyions sur une île déserte, et que nous n'ayions pas de table sous la main. Et nous voulons ce renseignement. Certains objecteront que, sur une île déserte, il n'y a pas non plus d'organisme de prêt. Passons sur ce détail.

La façon de faire consistera à essayer plusieurs montants de remboursements mensuels RM successifs en procèdant par APPROXIMATIONS SUCCESSIVES.

Soit NM le nombre de mois sur lequel devra courir l'emprunt.
On essaie une valeur de RM et on calcule la somme restant à liquider.
Si elle est négative c'est que RM a été choisi trop fort
Si elle est positive RM a été choisi trop faible.
On tâtonne. On se rapproche ainsi progressivement, le plus près possible de la valeur correcte.

Si l'emprunt porte sur 15 ans, ce qui est courant, le nombre de mois sera 360. Ce qui veut dire qu'il faudra, pour chaque choix de RM, faire trois cent soixante fois le calcul de S !

Si une douzaine d'essais sont nécéssaires, l'opération vire à l'interminable.

C'est là que l'ordinateur intervient.

L'exemple du taux d'intérêt est simple, et proche des préoccupations de l'homme de la rue. Quand on s'aventure dans la jungle du monde scientifique il ne faut pas chercher bien loin pour tomber sur des problèmes que seul l'ordinateur peut résoudre. La somme de calculs est tellement inimaginable que l'opérateur manuel mettrait des siècles à traiter le problème manuellement, sinon plus.

Brève remarque : en science, il n'y a pas de les calculs numériques. On effectue aussi souvent des calculs littéraux très importants. Par exemple dans le monde de la relativité.

L'ordinateur est aussi capable de manipuler des symboles codés. Des X, des Y etc...

On peut alors lui confier la tâche de tripoter des expressions littérales complexes, pour extraire telle ou telle forme EXPLICITE. Du style :

$$X = F (Y,Z,T,U,V,W)$$

On a déja utilisé avec succès l'ordinateur dans des tâches de ce genre. Par exemple en cherchant à vérifier des calculs littéraux. Et des erreurs ont ainsi pu être détectées.

Un ordinateur a ainsi pu refaire en quelques dizaines de minutes des calculs effectués par les hommes pendant des mois et des mois.

Je cite cette anecdote pour suggérer que l'ordinateur n'est pas ce BOUSTROPHEDON qu'on imagine.(De BOUS,le boeuf,et STROPHEDEIN le sillon).

Une nouvelle étape a été franchie.
Nous pourrions,si nous le désirions,programmer notre ORIC de manière à le rendre capable de calculer littérallement une dérivée d'une fonction quelconque.Mais c'est pousser le bouchon un peu loin pour un cours destiné en principe à des littéraires.

La façon dont l'ordinateur doit analyser une expression est assez passionante.Il est clair que,ce faisant,on progresse dans la voie des machines "intelligentes".

Le chemin est long.Et l'intelligence a de multiples visages. Ceela serait un peu compliqué d'expliquer comment se fait cette progressions.Refermons sagement cette parenthèse et revenons à notre algorithme de calcul d'emprunt.

Ici la solution est simple : l'ordinateur va effectuer le travail à notre place,beaucoup plus vite.

Tout cela commencera par une procédure de saisie de données:

```
10 CLS : REM EFFACAGE ECRAN
20  INPUT"TAUX D'ESCOMPTE EN % "; TE
30 INPUT"SOMME EMPRUNTEE:";S
40 INPUT"NOMBRE D'ANNEES";NA
```

Voilà,nous avons tous les renseignements nécéssaires.

Première chose à faire: convertir le taux d'escompte TE en taux de remboursement mensuel.

on rembourse TE/100 par an
donc TE/(100 * 12) par mois

écrivons :

50 A=TE / 1200

A devient le taux de remboursement mensuel (le alpha...)

Calculons le nombre de mois

60 NM = NA *12

RESOLUTION ITERATIVE :

Nous allons d'abord effectuer un calcul grossier du remboursement mensuel RM par tranches de 500 F.

Autrement dit on essayera d'encadrer la valeur cherchée de RM en testant successivement :

RM = 500

RM = 1000

RM = 1500

RM = 2000

etc.....

Le saut d'une valeut à l'autre s'appelera INCREMENT. Et l'opértion d'accroissement de RM s'appelle une INCREMENTATION

Nous allons créer une boucle de calcul, avec un ordre de retour en boucle et un test de sortie de boucle.

Pas d'affolement....

Dans un registre S1 je vais placer une copie de la valeur S de la somme empruntée, en faisant tout simplement

S1=S

Dans un registre IC je vais mettre l'INCREMENT (soit IC=500)

Le registre M permettre de compter ne nombre de mois. Sa valeur initiale sera M = 0 (initialisation de cette VARIABLE M).

Si on est capable de bien décrire le processus,on sera à même de programmer correctement.Quel est-il ?

Quelle est la valeur initiale à donner à RM dans cette itération. La valeur zéro ?
Ca marchera.Mais on peut gagner un temps précieux.On sait en effet que le temps de remboursement est infini si RM = S * A (là on se contente de payer l'intérêt sur la somme).
On écrira donc:

```
70  RM = S * A : REM VALEUR INITIALE DE RM
80  IC = 500
```

Puis écrivons :

```
100  S1=S : M= 0 : RM = RM + IC
```

On initialise les variables S1 et M et on incrémente RM.

Pas d'affolement.Cette écriture RM = RM + IC a quelquechose d'éminement absurde,algèbriquement parlant.Cela voudrait dire que IC est nul.On en 80 on a écrit qu'il ne l'était pas !

Le signe égal n'a pas en informatique le même sens qu'en algèbre.

Il se trouve que les informaticiens on utilisé ce signe égal dans ce langage de programmation.Il eut été préférable d'écrire:

$$RM \Leftarrow RM + IC$$

Ce qui signifie alors:

Dans le registre RM mettre la valeur initialement contenue (dans ce registre) PLUS la valeur présente dans le registre IC.

De même quand j'écris

```
IC = 500
```

il serait plus agréable de pouvoir écrire

IC ⇐ 5ØØ

Car nous savons que ceci signifie METTRE 5ØØ dans IC.

Bon... il se trouve qu'en informatique on a pris ce signe égal. Il faut s'en contenter. Mais j'espère que vous avez bien compris qu'il signifie METTRE DANS.

La quantité à "mettre dans" est APRES le signe égal et le registre intéressé est AVANT ce signe égal. D'où cette idée de flêche dirigée vers la gauche.

La ligne 100 représente donc une incrémentation. On accroit la valeur contenue dans ce registre RM de la valeur IC.

Ligne suivante :

11Ø S1 = S1 * (1 + A) - RM : M = M + 1

Ce qui signifie qu'on calcule le nouveau montant de la somme qui reste due après UN mois. Et on INCREMENTE le nombre de mois M

Encore une fois il eut été plus agréable de pouvoir écrire:

11Ø S1 ⇐ S1 * (1 + A) - RM : M ⇐ M + 1

On va réaliser cette opération jusqu'à ce que M atteigne le nombre NM de mois. Ainsi on écrira :

12Ø IF M <> NM THEN 11Ø

Autrement dit SI M EST DIFFERENT DE NM ALORS ALLER EN 11Ø

Ce TEST assurera une SORTIE DE BOUCLE AUTOMATIQUE dès que M atteindra la valeur entière NM.

Le retour vers la ligne 11Ø n'étant plus déclenché par le test on passera à la ligne suivante du programme, quelle qu'elle soit.

A ce stade,faisons tourner le programme.On entre le taux d'escompte,la somme S (par exemple T = 17 % et S = 200 000 F) emprunt sur cinq ans.

Après ce premier cycle de calcul,avec un remboursement mensuel égal à S ¥ A + 500 soit 2833 + 500 = 3333 F ,le reliquat d'emprunt S1 est de 153209.411 F

On peut s'en assurer en faisant

PRINT RM

Print S1.

Il faudra donc incrémenter de nouveau RM de la valeur IC. Et refaire un nouveau cycle de NM mois,pour obtenir une nouvelle valeur de S1,plus faible.

En fait il faut rééditer l'opération jusqu'à ce que S1 soit trouvé en valeur négative.

Ce qu'on pourra faire en écrivant

```
130  IF S1 > Ø  THEN 1ØØ
```

La sortie de boucle s'effectuera automatiquement lorsque S1 passera en négatif (la valeur de RM sera alors trop élevée, on aura trop remboursé).

Refaites l'essai.

Taux 17

Somme 200000

Années 5

Laisser infuser et faites print S1 ,vous obtenez: -33985.9532 F

La valeur de RM se situe donc entre

5333 F et 5333-500 F = 4833 F

(on trouve la valeur 5333 F en faisant print RM,bien sûr).

Réécrivons le petit bout de programme qui nous a donné ce résultat

```
10 CLS
20 INPUT"TAUX D'ESOMPTE EN %";TE
3Ø INPUT"SOMME EMPRUNTEE ";S
4Ø INPUT"NOMBRE D'ANNEES";NA
5Ø A=TE/12ØØ
6Ø NM=NA*12
7Ø RM=S*A
8Ø IC=5ØØ
1ØØ S1=S:M=Ø:RM=RM+IC
11Ø S1=S1*(1+A)-RM:M=M+1
12Ø IF M<>NM THEN 11Ø
13Ø IF S1>Ø THEN 1ØØ
```

on pourra complêter avec

140 RM = RM - IC : REM UN PAS EN ARRIERE

On pourra repartir alors de cette nouvelle valeur de RM et envisager un nouveau cycle de calcul avec un "pas" plus fin une valeur de IC dix fois plus petite : 50 F.

Il suffit de rééditer la séquence :

```
180 IC = 50
200 S1=S:M = 0 : RM = RM + IC
210  S1= S1 * ( 1 + A ) - RM : M = M + 1
220 IF M < > NM THEN 210
230 IF S1  >0 THEN 200
240 RM = RM - IC
```

Ce sont les mêmes lignes que précédemment, mais avec un IC dix fois plus faible.

On relance avec les mêmes valeurs

Taux 17
Somme 200000 (vingt bâtons)
Annéées 5
On obtient RM = 4933.33 F

Et la valeur exacte de RM est comprise cette fois entre
4933.33 F et 4933.33 + 50 = 4983.33 F
On a gagné dix fois en précision.
Reprenons tout cela avec cette nouvelle valeur de RM et un incrément IC égal à cinq francs (encore dix fois plus faible).
Il suffit de reproduire la même séquence opérative.

```
280  IC=5
300  S1=S:M=0:RM=RM+IC
310  S1=S1*(1+A)-RM:M=M+1
320  IF M<>NM THEN 310
330  IF S1>0 THEN 300
340  RM=RM-IC
     PRINT RM → 4968.33
```

Ceci va nous fournir une valeur de RM à cinq francs près.
On s'en contentera.

Exercice :

Rajoutez ce qu'il faut à ce programme pour que le remboursement mensuel soit palculé à un demi franc près.
Pour votre gouverne, toujours pour les mêmes données, vous trouverez que RM doit être compris entre 4970.33 F et 4970.83 F

Ce qui premettra de vérifier que vous avez opéré correctement.

Vous remarquerez que cette méthode d'itération permet d'atteindre une précision sans cesse accrue dans le calcul.
Si à chaque cycle on diminue dix fois le pas et si NC est le nombre de cycle de calcul, la précision sera en $1 / 10^{NC}$

EXERCICE :

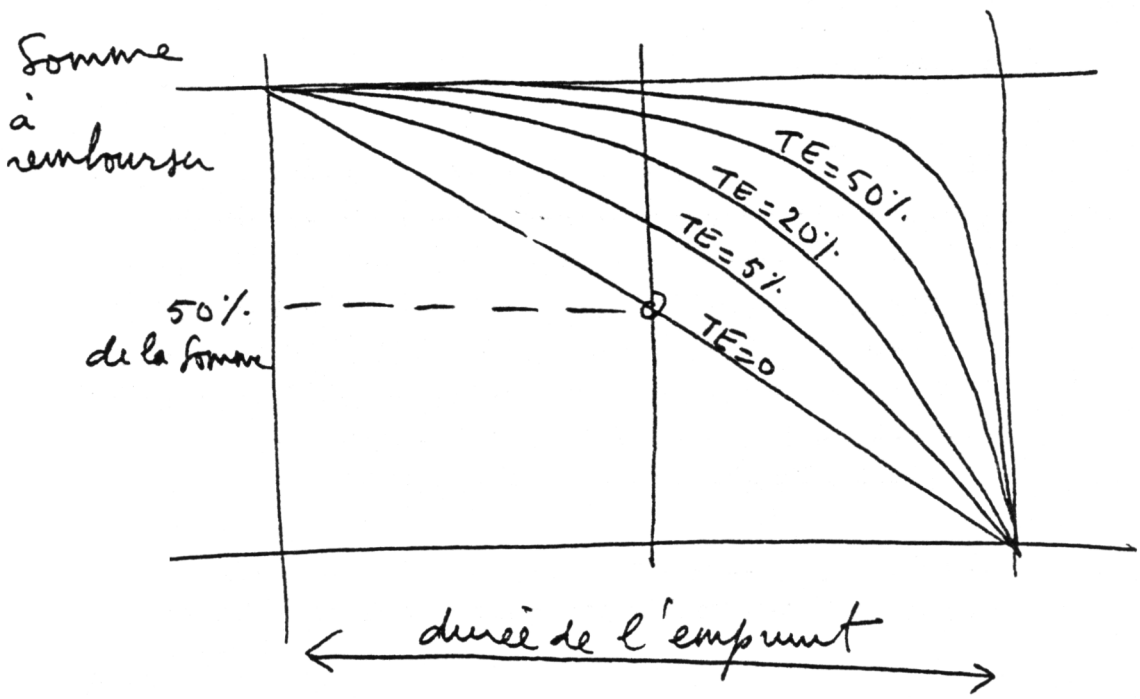
Vous connaissez donc la somme à rembourser mensuellement pour un taux d'escompte, une somme S et un nombre d'années d'emprunt donnés.

Ecrivez un programme qui fournit la somme à rembourser en cas de remboursement anticipé.

C'est le problème inverse et il est beaucoup plus simple.
Les "entrées" du problème sont :
La somme empruntée
Le taux d'escompte

Le remboursement mensuel,que vous venez de calculer

Le nombre d'années au bout duquel vous décidez de solder votre emprunt.
Si vous étudiez tout cela de près vous verrez que la somme à rembourser chute en fin de durée de l'emprunt,et que cette chute est d'autant plus accusée que le taux d'escompte est fort:



Si le taux d'scompte est nul,à chaque fois que vous payez RM, chaque mois,vous vous libérez de votre dette d'autant.Puisque vous n'avez aucun intérêt à payer.
Et on a tout simplement dans ce cas RM = S / NM

Situons nous à la moitié du temps de l'emprunt.Si vous remboursez alors prématurément et que le taux d'escompte est fort,vous vous apercevez que vous devez encore presque toute la somme.
Phénomène classique de l'emprunt.

Vous avez pu voir que votre ordinateur ne calculait pas tout cela instantanément.

Imaginez le travail si vous aviez du faire tout cela à la main.
Il manque quelquechose à ce programme. En effet, que faire si le remboursement mensuel est inférieur au 500 F, valeur initiale de IC.
Dans la ligne 70 on calcule la valeur plancher de RM.
Si celle ci est inférieure à 500 il suffira de démarrer à la ligne 180 en écrivant :
75 IF RM < 500 THEN 180

Et le tour est joué.

LE but de ce cours est de vous apprendre un peu de BASIC en vous montrant ce qu'on peut faire avec un ordinateur. Le chapitre précédent aurait pu s'intituler exemple de résolution de problème.
L'informatique est affaire d'astuce et d'habitude.

Il faut d'abord construire l'ALGORITHME permettant d'arriver au résultat (ici une méthode itérative). Puis bien écrire cet algorithme en décomposant chaque opération.

Enfin TRADUIRE tout cela en programme.

Il n'y a jamais UNE façon unique de répondre à une question posée.
Et en tout cas une façon unique d'écrire le programme.

L'essentiel est que cela marche et que cela ne prenne pas un temps excessif.

Faites l'expérience : prenez tout de suite une valeur de IC de 5F ou même de 0,5 F

Partez de la valeur de RM égale à S * A et incrémentez.
Bon courage.....

TOUT SUR LES BOUCLES

Une boucle,dans un programme,c'est une structure qui fait qu'une tâche est répètée un certain nombre de fois,soit identiquement, soit avec une variation discrète à chaque cycle d'un ou de plusieurs paramètres.

La boucle FOR NEXT

Dans ce type de boucle,le nombre de cycles est fixé à l'entrée.

On se donne un indice,qui n'est autre qu'un registre.Souvent les lettres I et J sont affectées à ce rôle.Mais cela peut être n'importe quoi.

```
FOR T =1 TO 56
```

Pour T variant de 1 à 56

On a vu que le point de départ de la boucle pouvait aussi être n'importe quoi.

```
FOR Z = 5 TO 9 par exemple.
```

On peut aussi partir d'une valeur négative de l'indice :

```
FOR I = -4 TO 9
```

La boucle FOR NEXT doit être marquée par un NEXT en fin de tâche.

```
FOR SF = 1 TO 12
 ( tâche )
NEXT SF   ( valeur suivante de l'indice SF )
```

Mais il n'est pas nécéssaire de spécifier l'indice.Et on peut

écrire :

```
FOR T = 3 TO 78
(tâche )
 NEXT
```

BOUCLES IMBRIQUEES

Des boucles FOR NEXT peuvent être imbriquées les unes dans les autres,comme ceci :

```
FOR I = 1 TO 10
 FOR J = 1 TO 5
  (Tache)
 NEXT J
NEXT I
```

Qu'on peut aussi écrire :

```
FOR I = 1 TO 10 : FOR J = 1 TO 5
(Tâche)
NEXT : NEXT
```

Dans toutes les écritures ci dessus,l'INCREMENT est automatiquement 1

STEP

Mais on peut spécifier un incrément différent,en écrivant:

```
10 FOR I=0 TO 50 STEP 5
20 PRINT I
30 NEXT I
```

A l'exécution:

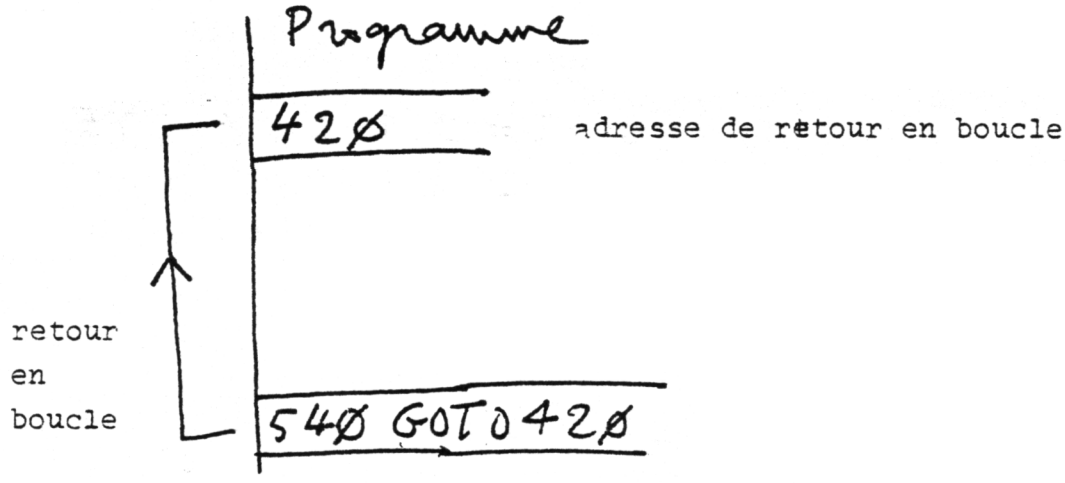
```
0
5
10
15
20
25
30
35
40
45
50
```

On peut mettre des INCREMENTS NEGATIFS :

```
10 FOR I=50 TO 0 STEP -5
20 PRINT I
30 NEXT I
```

BOUCLES AVEC DES GOTO :

Le shéma est le suivant :



Mais attention,une boucle,il faut en sortir !

Je vous propose le programme assez surréaliste suivant :

```
1Ø GOTO 1Ø
```

Faites RUN

Oh horreur,il ne se passe plus rien...
Vous pressez sur les touches,rien ne se passe.L'ordinateur est enfermé dans une tâche que Freud nommait la compulsion de répétition.Il réédite à l'infini une tâche.... vide.Même papa Sigmund n'aurait pas osé imaginer un cas aussi parfait.Le seul ordre de ce programme consiste à aller à la ligne 10 où l'ordre est précisément d'aller en ligne 1Ø ....

Que faire ?

Control C est l'ordre d'interruption prioritaire du programme. L'ordinateur réagit à ce qu'on appelle des CARACTERES DE CONTROLE.

La touche CTRL se comporte comme une autre touche SHFT.

Elle permet d'augmenter le nombre de caratères disponibles.
Ainsi Il existe CONTROL A
CONTROL B

etc....

Ces caratères de contrôle ont des fonctions qui seront examinées chemin faisant. Mais pour le moment retenons que CONTROL C stoppe les programmes en cours d'éxécution.

Il faut, comme pour la touche SHIFT commencer par presser la touche CTRL, maintenir cette pression, puis presser sur le caratère.
On note ces caratères

^{c}C ^{c}A ^{c}B etc..

On peut stopper un programme en composant ^{c}C et celui-ci s'arrête en cours d'exécution, en affichant le numéro de ligne qui était en train de s'exécuter. Essayons.

Nous voyons apparaitre BREAK IN 1Ø

Normal.... le programme ne comprenait qu'une seule ligne !
Essayez maintenant :

```
1Ø PRINT" BOUCLE INFINIE"
2Ø GOTO 1Ø
```

Et lancez.

Ca va tellement vite que vous n'avez même pas le temps de voir s'inscrire le mot.

Passons à

```
1Ø N=N+1
2Ø PRINT N
3Ø GOTO 1Ø
```

Ce type de programme vous permet d'apprécier la vitesse du retour en boucle.
Faites de nouveau Control C pour arrêter.

SORTIE DE BOUCLE

Une boucle construite avec un GOTO nécéssite un système de sortie de boucle. Ce qui se crée à l'aide d'un TEST.

```
10 N = N + 1
20 IF N = 50 THEN 40
30 GOTO 10
40 END
```

Lancez
Il ne se passe apparemment rien. Mais vous voyez réapparaitre le signe Ready qui vous montre que la tâche (vide) est terminée.
Cette suite d'instructions est équivalente à un

```
10 FOR I = 0 TO 50
20 NEXT I
```

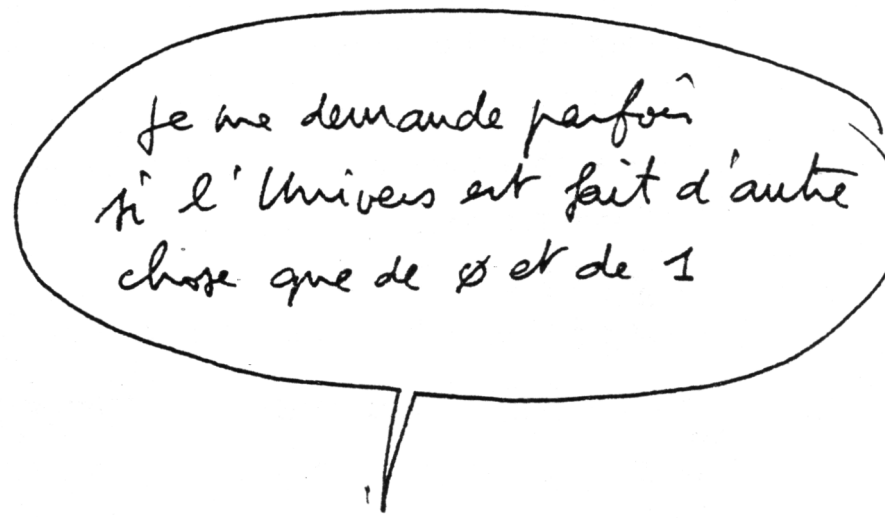
Ce qui illustre le fait que des instructions sont souvent des combinaisons d'instructions plus élémentaires. On peut simuler ce FOR NEXT avec un GOTO et un TEST de sortie de boucle.

Ceci nous ramène au modèle du MECCANO. Certaines pièces sont déja des assemblages de pièces plus simples. Un réducteur sera par exemple un assemblage d'axes et d'engrenages.

En fait c'est tout l'ordinateur qui marche ainsi. Au niveau le plus élémentaire, au niveau du langage machine, la seule chose que sait faire l'ordinateur c'est incrémenter d'une unité, ou décrémenter d'une unité, et comparer entre eux des registres qui n'ont que deux positions : 0 ou 1

C'est à partir de cette mécanique de base que l'on a tout imaginé. Un premier niveau est l'architecture du microprocesseur. Puis il y a le niveau du langage sur lequel on travaille.

En bas, tout en bas, c'est un remue ménage infernal de 1 et de Ø . Rien d'autre.



Revenons à nos boucles :

```
      1Ø N = 5
      2Ø N = N +5 : PRINT N
boucle 3Ø IF N = 5Ø THEN    5Ø  sortie de boucle
      4Ø GOTO 2Ø
      5Ø END
```

Ici l'incrément a la valeur 5.

Le test de sortie peut porter sur une inégalité.

exemple.

```
10  P = 45
20  P = 45 + 23.78
30 IF P >= 456.89 THEN 50
40 GOTO 20
50END
```

Avec ce système on peut évidemment créer des BOUCLES IMBRIQUEES.

Mais dans le travail précédent (pages 65 et suivantes)
On a créé une structurede boucle avec le TEST lui-même.

Rappelez vous :

```
8Ø IC=5ØØ
1ØØS1 = S : M=Ø : RM = RM + IC
11Ø S1 = S1 *(1+A)-RM : M=M+1
12Ø IF M<>NM THEN 11Ø
```

Dégageons dans ceci la structure essentielle.Il y a une incrémentation sur M et un test :

```
11Ø M = M+1
12Ø IF M<>NM THEN 11Ø
```

On a une valeur entière de NM.
M est incrémenté et la boucle continue de fonctionner tant que M n'est pas égal à NM.
Le reste,c'est la tâche.

En fait notre programme comportait deux boucles imbriquées.

Avec deux incrémentations,l'une portant sur M et l'autre sur RM

Voici le programme :

incrémentation

```
100 S1=S : M=0 RM=RM+IC
110 S1= S1*(1+A)-RM : M=M+1
120 IF M<>NM THEN 110
130 IF S1>0 THEN 100
```

test

En enlevant ce qui est tâche, il reste :

```
100
110
120 IF M<>NM THEN 110
130 IF S1>0 THEN 100
```

Ici ce sont des tests qui qui créent le retour en boucle et qui provoquent la sortie de boucle.

TEMPORISATION

Une boucle vide peut servir à créer un temps d'attente dans un programme. Par exemple :

```
FOR T = 1 TO 500
NEXT T
```

Cette suite peut figurer sous forme de sous programme :

```
10000 REM TEMPORISATION
10010 FOR T = 1 TO 500
10020 NEXT T
10999 RETURN
```

On pourra créer ce temps d'attente à une certaine étape d'un programme en appelant ce sous programme temporisation avec un GOSUB 10000

WAIT

(attendre)

L'ORIC possède une instruction de temporisation.
WAIT 50 créera une pause dans le programme équivalente à cinquante fois dix millièmes de sconde.
La encore nous voyons que les instructions comme WAIT ressemblent à des sous programmes codés. Nous pouvons simuler un WAIT

Ecrivons le sous programme WAIT :

```
10000 REM SOUS PROGRAMME  WAIT
10010 FOR Z = 1  TO T
10020 NEXT Z
10999 RETURN
```

Dans le programme, pour simuler cette instruction WAIT il faudrait faire :

```
T = 50
GOSUB 10000
```

Premier temps : on définit le temps d'attente
puis on appelle le programme de temporisation.
(Ceci dit, dans ce cas là il est possible que le temps d'attente ne soit pas le même, mais ce qui compte, c'est l'intention...)

EXERCICE :

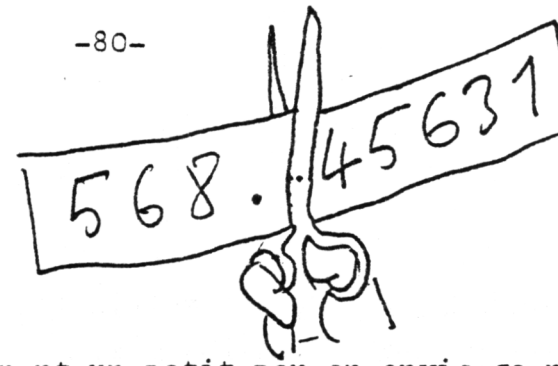
Utilisez l'instruction WAIT

Utilisez l'instruction PAPER (si vous avez la couleur)

Créez un programme qui change la couleur de l'écran

NB on rappelle qu'il y a huit couleurs disponibles indicées de 0 à 7.

INT



On va se reposer maintenant un petit peu en envisageant un programme plus simple que le difficile programme de calcul du remboursement mensuel d'emprunt.

Un numéro de sécurité sociale est une suite de chiffres qui réalisent un codage.
Le premier chiffre peut être un 1 ou un 2
1 signifie que le sujet est du sexe masculin
2 signifie que le sujet est du sexe féminin
Il n'y a pas d'autre nombre possible

Les deux chiffres suivants donnent l'année de naissance
Les deux chiffres suivants le mois de naissance
Les deux chiffres suivants le département où le sujet est né.
Les quatre chiffres qui suivent se réfèrent à un codage de la localité de la naissance et de du numéro d'entrée en scène du sujet le jour de sa naissance dans cette localité.
Nous voulons écrire un programme où l'entrée soit précisément ce numéro de sécu,ou plutôt ses sept premiers chiffres.Le programme va décoder ce numéro.

Cela commencera donc par

```
10 CLS : REM EFFACAGE ECRAN
20 INPUT "DONNEZ LES SEPT PREMIERS CHIFFRES DE
 VOTRE NUMERO DE SECU";NS
```

Comment pourrons nous isoler les chiffres ?

```
30 S=NS/1000000
40 S=INT(S)
50 IF S=1 THEN PRINT"VOUS ETES
DU SEXE MASCULIN": GOTO 80
60 IF S=2 THEN PRINT"VOUS ETES DU
SEXE FEMININ": GOTO 80
70 GOTO 10
80 REM
```

Dans la ligne 30 on divise NS par 1000000
A la ligne 40 on prend la partie entière de cette quantité.

Exemple: NS = 1370475

S= 1.370475

INT(S) donne 1

Deux tests produisent les interprètations souhaitées. Mais si le premier chiffre n'était ni un 1 ni un 2 la ligne 70 renvoit à l'ordre INPUT et rebelote.

Le premier chiffre ayant été analysé, comment extraire les deux suivants ?
Voilà la solution :
S est le registre qui contient donc ce premier chiffre (1 ou 2).

Si je reprends NS et que j'enlève S * 1000000 il me restera la suite.
Er j'écrirai :

```
80  S=NS - S * 1000000
```

Puis:

```
90 AN= S/ 10000
100 AN = INT ( AN )
```

Lancez, vérifiez, vous avez bien ces deux chiffres.

Vous allez pouvoir écrire :

```
110 PRINT"VOUS ETES NE EN 19";AN
```

et

```
120 S = S - AN * 10000
130M = INT ( S / 100)
```

Ce registre M représente, vérifiez le mois de naissance.
Il faudra alors écrire une suite de douze test destinés à créer un registre M $ contenant la chaine de caractère du mois.

```
140 IF M = 1 THEN M $ ="JANVIER" :GOTO 1000
150 IF M = 2 THEN m $ ="FEVRIER" / GOTO 1000
etc..etc...
```

et:

```
1000 PRINT"AU MOIS DE "; M $
```

EXERCICE : terminez ce programme et débrouillez vous pour isoler les deux derniers chiffres.Puis créez les tests qui qui définiront _ D $, le département de naissance.

EDITEUR DE L'ORIC

Quand vous travaillez avec votre ordinateur,vous écrivez des textes sur l'écran,et les erreurs sont sans doute nombreuses. Nous avons vu que la touche DEL permettait d'annuler un caractère erroné,ou même,en rééditant la frappe de cette touche,plusieurs.

CONTROL X

Si vous avez tapé quelquechose de vraiment déconnant,un ordre que vous voulez annuler,composez Ctrl X
Le signe \ apparaitra et tous les caratères frappés seront "oubliés" ,c'est à dire qu'ils ne seront pas envoyés dans l'interprèteur BASIC.

C'est le "AUTANT POUR MOI".

Au bout d'un certain temps vous finirez par manipuler des programmes qui ne pourront plus être affichés d'un coup sur l'écran. Ils seront trop longs.

Quand vous faites LIST le programme se met à défiler. Vous pouvez arrêter ce défilement en pressant simplement sur la barre d'espacement. Une nouvelle pression relancera ce défilement. Très commode.....
(Sur l'APPLE II la commande équivalente est Ctrl S).
Vous pouvez stopper définitivement ce défilement en faisant Ctrl C

MODIFIER UNE LIGNE DU PROGRAMME

Jusqu'ici pour modifier une ligne nous nous sommes contentés de la retaper complêtement.

Noter qu'une ligne de l'ORIC ne saurait excèder 78 caractères.

Voici une ligne erronée :

40 AN = INT(AN) ; GOTO 1000

Vous avez tapé un point virgule au lieu de deux points.
Commencez par utiliser les touches porteuses de flêches pour amener le curseur immédiatement à gauche de la ligne, comme ceci:

■40 AN = INT(AN) ; GOTO 1000

La commande Ctrl A va entrainer le déplacement du curseur vers la droite. Si vous maintenez la touche Ctrl pressée et que vous pressiez plusieurs fois sur la touche A, vous obtiendrez autant de déplacements succesifs vers la droite.
Vous pouvez aussi bien maintenair l'ensemble Ctrl A pressé, auquel cas, après un temps d'attente, le cirseur va partir tout seul. Et avec un peu de chance et d'adresse vous pourrez le faire atterrir pile sur l'endroit litigieux.

En agissant ainsi, vous RECOPIEZ la ligne.
Quand vous êtes sur le caractère à modifier, pressez la bonne touche. Mais attention, il faut recopier TOUTE LA LIGNE, c'est à dire finir de la balayer avec Ctrl A.

Voilà.

Ceci est particulier à l'ORIC. Sur l'APPLE II la procédure est différente.

CLS vide l'écran. Mais il subsiste le message Ready.

Ctrl L vide complêtement l'écran.

TRON TROFF

L'instruction TRON, incluse dans un programme, va provoquer l'affichage , entre deux crochets, du numéro de ligne qui est effectué. TROFF annulle cet affichage. Cela peut être commode lors d'une recherche d'erreur.

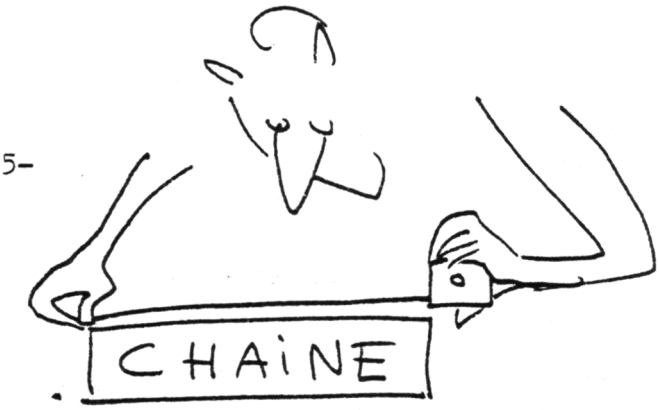
Voilà l'essentiel de l'EDITEUR de l'ORIC.

Nous allons passer à quelquechose de très intérssant : les instructions de

TRAITEMENT DE TEXTE

Nous avons vu qu'un ordinateur pouvait opérer sur deux objets différents. Des nombres, placés dans des registres numériques. Et des CHAINES DE CARATERES. Placées dans des registres suivis du signe "dollar" $.

LEN



Cette instruction LEN permet de calculer la longueur de la chaine, le nombre de caractères qu'elle contient.

```
10 CLS
20 INPUT"DONNEZ VOTRE CHAINE";C $
30 L = LEN ( C $ )
40 PRINT"CETTE CHAINE FAIT ";L;" CARACTERES"
```

LEFT $

Cette instruction permet de prélever un certain nombre de caratères à partir de la gauche.

```
5 CLS
10 INPUT"DONNER CHAINE";C $
20 ?:?"COMBIEN DE CARACTERES VOULEZ-VOUS"
30 ?:?  A PARTIR DE LA GAUCHE"
40 ? : INPUT N
50 G$ = LEFT$(C$,N)
60 PRINT G$
```

RIGHT $

Réalise la même opération,mais à partir de la droite.

<u>Exercice</u> : Modifiez ce court programme avec R$= RIGHT$(C$,N)

MID$

Cette instruction permet de créer une SOUS-CHAINE de caratères piquée au beau milieu d'une autre chaine.

Soit la chaine : les pantoufles sont cheres

Le registre M$ = MID$(C$,5,10) contient PANTOUFLES

Entre parenthèses d'abord la chaine à traiter
puis la cote du caractère de début de sous chaine
puis la longueur de cette sous chaine.

```
5 CLS
10 INPUT"DONNEZ CHAINE";C$
20?:?"POUR CREER UNE SOUS CHAINE,DONNEZ LE NUMERO"
30 ?:?"DU CARATERE DE DEBUT DE SOUS-CHAINE"
40 INPUT N
50 ?:?3DONNEZ ENSUITE LA LONGUEUR SOUS-CHAINE"
60 INPUT L
70  M$ = MID$(C$,N,L)
80 ?:?3 CETTE SOUS CHAINE EST:"
90 ?:? M$
```

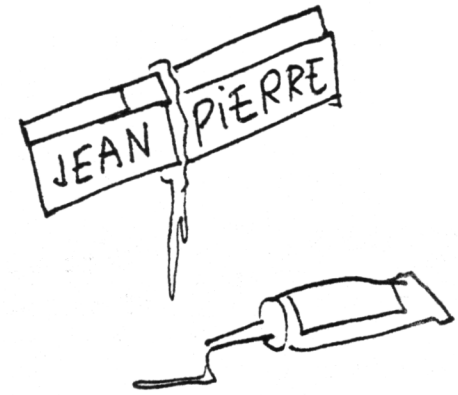
ATTENTION LES BLANCS SONT DES CARATERES COMME LES AUTRES.

Nous allons créer un programme classique qui,à partir d'une phrase, isole les mots les uns après les autres.

Cela commencera par

```
5 CLS
10 ?"DONNEZ UNE PHRASE"
20 INPUT P$
```

ISOLER DES MOTS



CONCATÉNATION:

On peut souder deux chaines de caractères entre elles,pour former une nouvelle chaine qui soit la REUNION des deux.

Il suffit de faire:

```
C$ = C1$ + C2$    tout simplement.
```

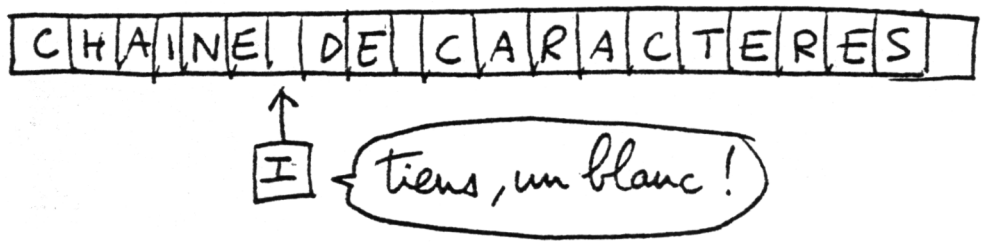
Nous allons ici ajouter à cette chaine P$ un caractère blanc.

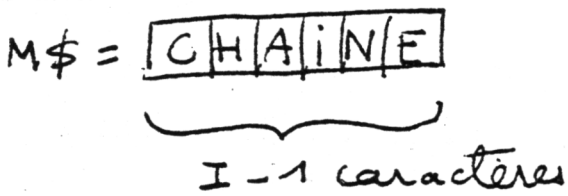
```
30 P$ = P$ + " "
```

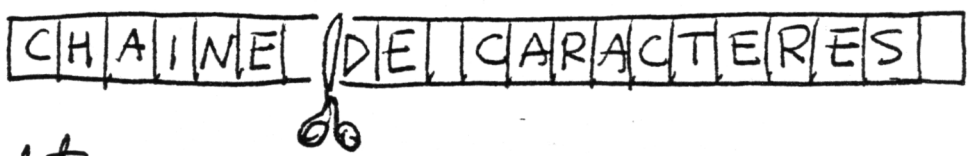
Ceci trouvera plus loin sa justification.

La façon d'isoler les mots est simple.Nous allons parcourir la chaine,caratère après caratère,jusqu'à ce que nous trouvions soit un blanc,soit une apostrophe.Nous nous arrèterons alors et nous prélèverons ce mot avec un LEFT$
Puis nous remplacerons P$ par ce qui reste,à droite,à l'aide d'un RIGHT$.Et nous reprendrons cette opération autant de fois qu'il le faudra.
On s'arrètera quand on arrivera au bout de la chaine P$.

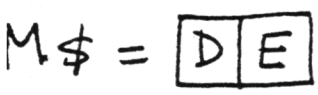
La longueur de la chaine est L
Nous allons parcourir la chaine selon un indice I :

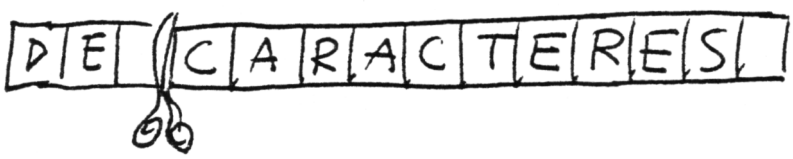


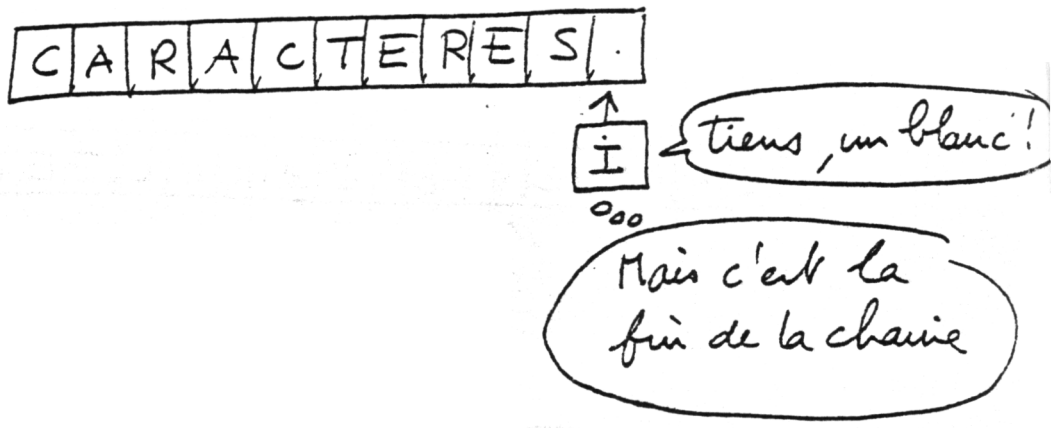




rebelote :







Traduisons ces petits dessins en programme.

```
50 I = 0 : REM MISE A ZERO COMPTEUR CARACTERES
60 I = I + 1  : REM INCREMENTATION
70 C$ = MID$(P$,I,1) : REM ISOLER UN CARACTERE
80 IF C$ <> " " AND  C$ <>"'" THEN 60

90 M$ = LEFT$(P$,I-1)

100 PRINT M$ : L = LEN(P$)
110 IF L = I THEN END
120 P$ = RIGHT$(P$,L-I) : GOTO 50
```

Et le tour est joué.

En 60 on incrémente,on parcourt la chaine.
70 on isole le I° caractère
80 Tant que ce caractère n'est ni un blanc ni une apostrophe on continue.
Mais si ce test est négatif,c'est qu'on est à la fin d'un mot.
Le mot est alors créé,en prenant les I-1 premiers caratères de la chaine.On imprime le mot.
On calcule la longueur de la chaine P$.Si L est égal à I c'est qu'on est arrivé à la fin de la phrase.Alors on s'érrète.
Sinon on va isoler ce qui reste à droite,c'est à dire les L-I caratères de droite.Et on retourne à la ligne 50.

Faisons un essai.

```
RUN
Phrase :
J'AIME LE JAMBON ET LA SAUCISSE
J
AIME
LE
JAMBON
ET
LA
SAUCISSE
```

Ca marche....

VARIABLES INDICÉES

Nous avons écrit les mots successifs sur l'écran. Mais nous ne les avons pas mémorisés. Si ce bout de programme devait être le préambule à un traitement de la phrase, nous ne serions pas plus avancés. Il nous faudrait créer un FICHIER de ces mots. Un fichier

M$(I)

Voici le programme qui va constituer ce fichier des mots de la phrase :

```
5CLS
20PRINT "VOTRE PHRASE" : PRINT
30 INPUT P$ : PRINT
40 P$=P$+" "
50 J=0
60 J=J+1
70 I=0
80 I=I+1
90 C$=MID$(P$,I,1)
100 IF C$<>" " AND C$<>","THEN 80
110 M$(J)=LEFT$(P$,I-1)
120 PRINT M$(J)
130 L=LEN (P$)
140 IF L=I THEN 170
150 P$=RIGHT$(P$,L-I)
160 GOTO 60
170 FOR K = 1 TO J
180 PRINT M$(K) : PRINT
190 NEXT K
200 END
```

En 50 on met à zéro le compteur de mots
En 60 on incrémente ce compteur
En 80 on commence le parcours de la chaine qu'on poursuivra tant que (ligne 100) le caractère isolé n'est pas un blanc ou une apostrophe.
Sinon en 110 on crée le J° mot.
Etc...
En 170,en fin de programme,on affichera tous les mots isolés.

FICHIERS

On peut donc,dans la mémoire de l'ordinateur,organiser les données en créant la possibilité de les repérer avec un ou plusieurs indices.

A$ (I) B (I)

C$ (I,J) D (I,J)

EL$ (I,J,K) F1 (I,J,K)

Quand le répérage se fait avec plusieurs indices on appelle ceci des TABLEAUX.

I \ J			
1	D(1,1)	D(1,2)	D(1,3)
2	D(2,1)	D(2,2)	D(2,3)
3	D(3,1)	D(3,2)	D(3,3)
4	D(4,1)	D(4,2)	D(4,3)

DIM

On peut ainsi créer des fichiers,des tableaux à indices. Tous les noms possibles de variables peuvent être indicés.

A(I) CG(J) P1$(K) etc....

A priori la machine tolère des indices allant de Ø (qui compte comme indice possible))à 9.

Au delà,problèmes....

Faites ? S(12)

Vous obtenez

? BAD SUBSCRIPT ERROR

Erreur d'indice.

Pour manipuler des indices supérieurs à 9 il faut créer ces fichiers en mémoire à l'aide d'ordres de dimensionnement DIM.

Supposns que je veuille manipuler un fichier S(I) où I peut aller de zéro à 32Ø.

Simple,je dois mettre en tête de programme :

DIM S(32Ø)

ATTENTION,on ne peut lire deux fois un ordre dim en effectuant un programme. Si vous mettez par exemple un DIM dans une boucle vous obtiendrez le message d'erreur :

? REDIM'D ARRAY ERROR

Si vous avez plusieurs tableaux vous pouvez simplement mettre :

1Ø DIM S(32Ø), P$(12), SC(4Ø)

PROCÉDURE DE RÉPONSE PAR OUI/NON

Ce cours n'est pas destiné qu'aux étudiants littéraires,ou aux maniaques de l'intelligence artificielle.Il va comprendre une petite initiation à la gestion.
Il est bon d'envisager,ou la procédure de choix appelée MENU qu'on a vu plus haut,les procédires avec réponse oui ou non.

On peut supposer que,quelquepart dans le programme,on.posera une question:

```
1150  INPUT"VOULEZ-VOUS EFFECTUER UNE AUTRE SAISIE ? ";R$
```

La réponse étant évidemment consignée dans le registre R$.
On souhaite qu'elle soit oui ou non.

La répinse OUI renvoit par exemple à l'adresse 1000.E, la réponse NON à la suite du programme.Mais il faut éliminer une réponse autre que non.On écrira :

NOT

```
1160 IF R$="OUI" THEN 1000
1170 IF NOT(R$="NON") THEN 1150
```

Cette dernière ligne signifiant :

Si R$ N'EST PAS "NON".

Mais cela peut sembler assez long de taper les trois lettres d'une réponse OUI ou NON. On aimerait se contenter de O et de N. Mais que faire si certains utilisateurs répondent OUI et d'autres simplement O ?

On écrira, en utilisant l'instruction de traitement de chaine LEFT $:

```
1160 IF LEFT$(R$,1) = "O" THEN 1000

1170 IF NOT(LEFT$(R$,1) = "N") THEN 1150
```

Ce qui veut dire :

SI LA PREMIERE LETTRE DE LA REPONSE EST UN O ALLER EN 1000
SI LA PREMIERE LETTRE DE LA REPONSE N'EST PAS UN N ALLER EN 1150

C'est la solution classique.

A partir de là nous allons commencer à constituer un programme de gestion. Les étudiants littéraires devront lire cette partie. Car c'est un bon apprentissage de l'informagique, qui n'est du début à la fin qu'une perpétuelle gestion de données.

ELEMENTS DE GESTION

Dans ce qui va suivre on va donner un programme de gestion de comptes .Programme qui sera gèré par un MENU présentant plusieurs choix :

CREER UN FICHIER-CLIENTS
COMPLETER UN FICHIER-CLIENTS
CONSULTER UN FICHIER-CLIENTS
METTRE A JOUR UN COMPTE-CLIENT
RELANCER LES IMPAYES
ANALYSER LES COMPTES-CLIENTS

ChaQUE CHOIX RENVOYANT & UN SOUS PROGRAMME.

On écrira d'abord le programme MENU,puis ces sous programmes.

Nous allons manipuler un certain nombre de VARIABLES.Dans des fichiers.

N$ va représenter le nom des clients
R$ sa rue
AR$ l'arrondissement
V$ la ville où il réside
P$ le pays
CC son compte-client. (variable numérique. le reste étant des chaînes de caractères).

Combien de clients pouvons nous prendre en charge avec un ORIC ? Tout commencera par un ordre de réservation de places mémoire.

On peut effectuer ce dimensionnement en commande directe,sans passer par un programme.Mais le programme sera une solution plus commode pour répondre vite à la question que nous nous posons. Ecrivons;

```
1Ø REM PROGRAMME DE GESTION DE COMPTES-CLIENTS
2Ø REM RESERVATION DES PLACES EN MEMOIRE
25 INPUTN
3Ø DIM N$(N),R$(N),AR$(N),V$(N),P$(N),CC$(N)
```

Faites RUN,avec différentes valeurs de N (remarquez que l'ordinateur n'est pas gène de la présence de N et de N$,qui sont traités comme deux objets différents.)

Faites N=1ØØ

pas de ppoblème

Faites N=2ØØØ

Message : ? OUT OF MEMORY ERROR IN 2Ø

Eh oui,la capacité mémoire n'est pas infinie.Ici vous n'utilisez l'ORIC que comme classeur.Vous lui demandez de se transformer en une suite de six fichiers,les cinq premiers servant à stocker des chaines de caratères et le dernier des quantitéé numériques.

Par approximations successives vous verrez que votre mémoire se trouve remplie pour N = 1961.

Est-ce à dire qu'on pourrait TRAITER 1961 clients dans ce programme.Non.Car il faut ménager une place pour le programme. Données et programme sont stockés indifféremment dans ma MEMOIRE UNIQUE de cet ordinateur.Il n'y a pas comme dans certaines calculett des secteurs fixés à l'avance et intitulés : MEMOIRE DONNEES et MEMOIRE PROGRAMME.

C'est ce qui donne le maximum de souplesse car alors on peut choisir entre

gros programme travaillant sur peu de données

programme tenant peu de place en mémoire, mais beaucoup de données.

Reprenons ce programme. Enlevons cette ligne provisoire 25 et écrivons :

```
10 REM Programme de gestion de comptes clients
20 Rem Reservation de Place en memoire
30 DIM N$(100), R$(100), AR$(100), V$(100),
P$(100), CC(100)
```

Autrement dit cent clients.

5 a la fin de la constitition de cd programme vous pourrez essayer de monter les DIL pour voir quel nombre de clients vous pourriez gèrer avec. Je vous laisse ce travail à titre d'exercice).

Ecrins le programme MENU.

On veut, sur l' écran :

```
VOULEZ-VOUS
1-CREER UN FICHIER CLIENTS
2-COMPLETER UN FICHIER CLIENTS
3-CONSULTER UN FICHIER-CLIENTS
4-METTRE A JOUR UN COMPTE-CLIENT
5-RELANCER LES IMPAYES
6-ANALYSER LES COMPTES-CLIENTS
9-SORTIR DU PROGRAMME
VOTRE CHOIX ?
```

MENU :

```
100 CLS
110 REM PROGRAMME MENU
120 PRINT"VOULEZ-VOUS:": PRINT
130 PRINT"1- CREER UN FICHIER CLIENTS": PRINT
140 PRINT"2-COMPLETER FICHIER-CLIENTS": PRINT
150 PRINT"3-CONSULTER FICHIER.CLIENTS": PRINT
160 PRINT"4-METTRE A JOUR COMPTE-CLIENT":
170 PRINT"5-RELANCER IMPAYES": PRINT
180 PRINT"6-ANALYSER COMPTES-CLIENTS": PRIN
190 PRINT"9-SORTIR DU PROGRAMME": PRINT
200 INPUT"VOTRE CHOIX";C: PRINT
210 IF C=9 THEN END
220 ON C GOSUB 1000,2000,3000,4000,5000,6000
290 REM RETOUR MENU
300 GOTO 100
```

On remarquera que la sortie de programme est l'option 9.
Ceci laisse de la place dans ce menu pour d'autres options 7 et 8 qui renverraient à des sous programmes 7ØØØ et 8ØØØ ,non encore existants.
Seule l'option 9 permet de sortir du programme.On est enefermé dans cette boucle du MENU,ou l'on cercle.L'option 9 assurant la SORTIE DE BOUCLE.

Evidemment on a pas tenu compte des réponses aberrantes.
Ce qu'on pourra faire en écravant une ligne

```
215 IF C<>1 AND C<>2 AND C<>3 AND C<>4
AND C<>5 AND C<>6 AND C<>7 AND
C<>8 AND C<>9 THEN 2ØØ
```

Toute autre réponse que 1,2,3,4,5,6,7,8,9 renvoit au questionnement 2ØØ
Mais que se passera-t-il par exemple si le sujet répond par la lettre O,ou par uncaractère.
La saisie ne se fera pas
Message ORIC :
? REDO FROM START
VOTRE CHOIX ?
Rebelote.Le problèmes est bien verouillé.
Bon,plus qu'à écrire les sous-programmes.

```
1ØØØREM CONSTITUTION FICHIER CLIENTS
1Ø1Ø REM LISTE CLIENTS + LEUR COMPTES
1Ø2Ø CLS
1Ø3Ø I = Ø : REM INITIALISER VARIABLE I
```

```
1040 I = I + 1 : REM  INCREMENTER I
1050 INPUT"NOM CLIENT "; N$(I): PRINT
1060 PRINT"SON ADRESSE ": PRINT
1070 INPUT"NOM RUE "; R$(I) : PRINT
1080 INPUT"NUMERO"; N$ : PRINT
1090 REM CONCATENATION :
1100 R$(I) = N$ + " RUE " + R$(I)
```

Voilà qui donne une standardisation de cette saisie.
Il est inutile de créer un fichier pour les xxxx numéros dans les rues. On effectue donc une saisie provisoire du numéro, considéré comme un caractère, dans le registre N$. Puis on saisit la rue dans R$(I). Enfin on transforme le registre R$(I) par CONCATENATION (CATENA veut dire en latin chaine).

Si la rue est CLEMENCEAU et le numéro 6 après concaténation le registre R$(I) contient la chaine 6 RUE CLEMENCEAU

Et le tour est joué.

```
1110 INPUT"VILLE "; V$(I) : PRINT
1115 INPUT"ARRONDISSEMENT: "; AR$(I): PRINT
1120 INPUT" PAYS"; P$(I) : PRINT
1130 INPUT"COMPTE"; CC(I)
```

La saisie client est terminée. Mais que se passe-t-il si on a fait une erreur ? on risque de planter toute l'affaire. Il faut donc introduire une séquence de vérification :

```
11400 CLS
11500 PRINT"IL S'AGIT BIEN DE:"PRINT
11600 PRINT N$(I) : PRINT
11700 PRINT R$(I) : PRINT
11800 PRINT"A "; V$(I);"- "; P$(J):PRINT
11850 PRINT"ARRONDISSEMENT";AR$(I):PRINT
11900 PRINT"COMPTE :"; CC(I);"  F":PRINT
12000 INPUT"CORRECT";R$ : PRINT
12100 IF LEFT$(R$,1)="N" THEN CLS:
GOTO 10500
```

Si la réponse est NON ou plutôt si la première lettre de cette réponse est N alors on retourne en 1Ø5Ø après effaçage écran,et rebelote.
Mais l'indice I n'est pas modifié.Ce qui fait que ce sera toujours le même client.Les valeurs saisies N$(I) predront la place des précédentes et la correction sera automatiquement effectuée.
Si la xx saisie a été effectuée correctement,suite du sous programme

```
12200 CLS
12300 INPUT"UN AUTRE CLIENT";R$
12400 IF LEFT$(R$,1)="O" THEN CLS:
GOTO 10400
```

On se rebranche alors sur la ligne réalisant l'incrémentation sur I

Si on ne veut pas d'autre client,suite du sous programme:

On stocke dans le registre NC le nombre de clients du fichier.

```
1250 NC=I :CLS
1999 RETURN
```

Ce programme de constitution de fichier clients ne sera exécuté qu'une fois.Et on retournera au menu.Pour introduire un nouveau client on utilisera l'option 2

```
2000 REM COMPLETER FICHIER-CLIENTS
2010 CLS
2020 PRINT"CE FICHIER CONTIENT";NC";"CLIENTS
2030 PRINT: NC = NC+1
2040 PRINT"NOUVEAU CLIENT: ":PRINT

2050 INPUT"NOM ";N$(NC):PRINT
```

Ici,surprise : l'indice a changé !

Alors ? ça n'est plus le même fichier ?

Pas du tout.Ce qui compte,ça n'est pas la ou les lettres qui désignent l'indice,mais la valeur NUMERIQUE qui se trouve dans ce registre.

Si on stocke une chaine de Caractères dans N$(I) ou N$(NC) ou N$(K) ou N$(n'importe quoi...) si les contenus des registres sont les mêmes,cela atterrira dans le même élément de ce fichier.

```
2060 PRINT"ADRESSE" :PRINT
2070 INPUT"NOM DE LA RUE"; R$(NC) :PRINT
2075 INPUT"NUMERO DANS LA RUE"; N$ :PRINT
2080 R$(NC) = N$ + " RUE " + R$(NC)
2085 INPUT "VILLE"; V$(NC) : PRINT
2090 INPUT" ARRONDISSEMENT"; AR$(NC) :PRINT
2100 INPUT"PAYS"; P$(NC) :PRINT
2105 INPUT"COMPTE="; CC(NC) :PRINT
2110 PRINT"IL S'AGIT BIEN DE" :PRINT
2120 PRINT N$(NC)
2130 PRINT R$(NC)
2140 PRINT V$(NC);"- "; AR$ (NC)
2150 PRINT P$(NC)
2160 PRINT "COMPTE :" ; CC(NC);" F":PRINT
2170 INPUT"CORRECT" ;R$
2180 IF LEFT$ (R$,1) ="O" THEN 2210
2190 CLS : GOTO 2040
2210 INPUT"UN AUTRE CLIENT" ;R$ :PRINT
2220 IF LEFT$(R$,1)="O"THEN 2030
2999 RETURN
```

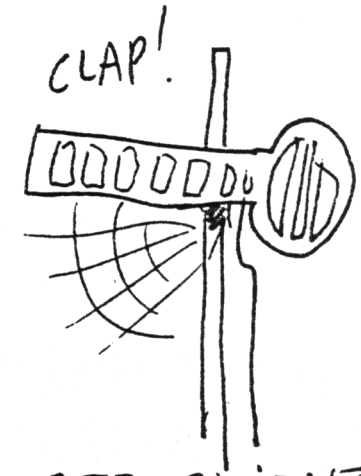
Passons à l'option 3

```
3000REM CONSULTATION FICHIER.CLIENTS
3010 CLS
3020PRINT"VOULEZ-VOUS :": PRINT
3030 PRINT"EXAMINER UN COMPTE
CONNAISSANT LE :": PRINT
3040 PRINT"1-NOM DU CLIENT": PRINT
3050 PRINT"2-NUMERO DU CLIENT":PRINT
3060 INPUT"VOTRE CHOIX"; C
3070 IF C<>1 AND C<>2 THENCLS :
GOTO 3020
3080 IF C = 1 THEN GOSUB 10000
3090 IF C = 2 THEN GOSUB 11000
3999 RETURN.
```

Voici un sous-programme qui contient lui-même un SOUS-MENU à deux options,renvoyant à deux autres sous programmes 10000 et 11000

On les écrira tout à l'heure...

En fait cette sous-tâche constituant à rechercher le client,soit par son numéro,soit à l'aide de son nom,reviendra plusieurs fois dans les options suivantes.Raison de plus pour renvoyer cela dans des sous programmes.



Routine de mise à jour de compte client :

FLAG

```
4000 REM MISE A JOUR COMPTE-CLIENT
4010 CLS
4020 PRINT"1- VOUS CONNAISSEZ LE NOM
DU CLIENT" :PRINT
4030 PRINT"2- VOUS CONNAISSEZ SON
NUMERO DE COMPTE" :PRINT
4040 FL=0 : REM MISE A ZERO FLAG
4050 INPUT"VOTRE CHOIX";C
4060 IF C<>1 AND C<>2 THEN 4000
4070 IF C=1 THEN GOSUB 10000
4080 IF C=2 THEN GOSUB 11000
```

Qu'allons nous mettre dans ces sous programmes 10000 et 11000 ?

Nous allons nous en préoccupper maintenant.
On voit donc qu'on peut composer les sous programmes dans un ordre quelconque (à condition que l'ensemble reste cohérent !)
L'ordinateur met tout cela dans l'ordre.

```
10000REM EXAMEN DE COMPTE,NOMINATIF
10010 CLS
10020 INPUT"NOM DU CLIENT"; N$ :PRINT
10030 I = 0
10040 I=I + 1 : REM INCREMENTATION
10050 IF N$ = N$(I) THEN FL = 1 :
PRINT"COMPTE = "; CC(I);" F" :WAIT300
GOTO 10999
10060 IF I = NC THEN PRINT"NOM HORS
FICHIER" : WAIT 300 : GOTO 10999
10070 GOTO 10040
10999 RETURN
```

Cette séquence mérite quelques éclaircissements.

En 10030 on met I à zéro (initialisation de la variable).

Suit une "boucle d'exploration de fichier". Avec une incrémentation.

On recherche le nom du client en parcourant tout le fichier N$(I).

en 10050 un test. Si le nom N$ est trouvé, c'est à dire s'il s'identifie à un élément N$(I) alors le FLAG F (qui n'est qu'un registre qulconque servant de FLAG) est positionné à 1. On imprime le renseignement demandé, avec un ordre de temporisation WAIT. Le temps d'affichage doit être suffisant pour permettre à l'opérateur de noter, le cas échéant.

Si le nom a été trouvé, inutile de poursuivre et on renvoit en 10999 ,au RETURN de la fin de sous tâche recherche du client.

Supposons que le nom ne soit pas dans le fichier. La boucle va se poursuivre jusqu'à ce que I = NC. Le fichier a alors été exploré complêtement. On donne alors l'information

NOM HORS FICHIER

Et le FLAG N'AURA PAS ETE MIS A 1 !

Qu'est-ce donc qu'un FLAG ?

C'est une mémoire de la chose faite.
En 4Ø4Ø on a initialisé le FLAG à zéro. On part donc du fait qu'on ne connait pas le nom du client.

Si le client est trouvé, le FLAG sera mis à 1. On pourra se servir de cette information par la suite. Inutile d'opérer sur un compte si on sait pertinement que le client N'EST PAS dans le fichier.

Passons à la recherche selon le numéro du client.

```
11ØØØ REM CONSULTATION FICHIER PAR NUMERO
11Ø1Ø CLS
11Ø2Ø INPUT"NUMERO DU CLIENT";I:PRINT
11Ø3Ø IF I>NC THEN PRINT"NUMERO HORS
FICHIER":WAIT 3ØØ:GOTO 11999
11Ø4Ø PRINT"NOM DU CLIENT";N$(I):PRINT
11Ø5Ø PRINT"COMPTE :";CC(I):" F" :
WAIT 3ØØ:FL=1
11999 RETURN
```

Si le client a été trouvé (si son numéro n'excède pas le nombre de client du fichier : NC) le flag est "armé".

Revenons à notre sous-programme 4ØØØ

```
4Ø6Ø IF FL=Ø THEN PRINT "ERREUR":
WAIT 1ØØ : GOTO 4999
```

Le client n'a pas été trouvé (le flag est resté nul)

```
4Ø7Ø PRINT"IL S'AGIT BIEN DE :" : PRINT
4Ø8Ø PRINT N$(I) : PRINT
4Ø9Ø PRINT"COMPTE :" ; CC(I) ;" F":PRIN
```

Le registre I contient le numéro du client dans les deux cas de figure. Soit (routine 11ØØØ) on l'a donné au clavier. Soit (routine 1ØØØØ) il a été défini dans la boucle d'exploration.

```
41ØØ PRINT"INDIQUEZ L'OPERATION, POUR
UN REGLEMENT" :PRINT
411Ø PRINT"FAITES PRECEDER D'UN +" :PRINT
412Ø PRINT"POUR UN DEBIT ,D'UN -" : PRINT
413Ø INPUT"MONTANT DE L'OPERATION"; OP:PRINT
414Ø CC (I) = CC(I) + OP
415Ø PRINT"NOUVEL ETAT DE COMPTE";
CC(I) ;" F" : WAIT 3ØØ
4999 RETURN
```

GET

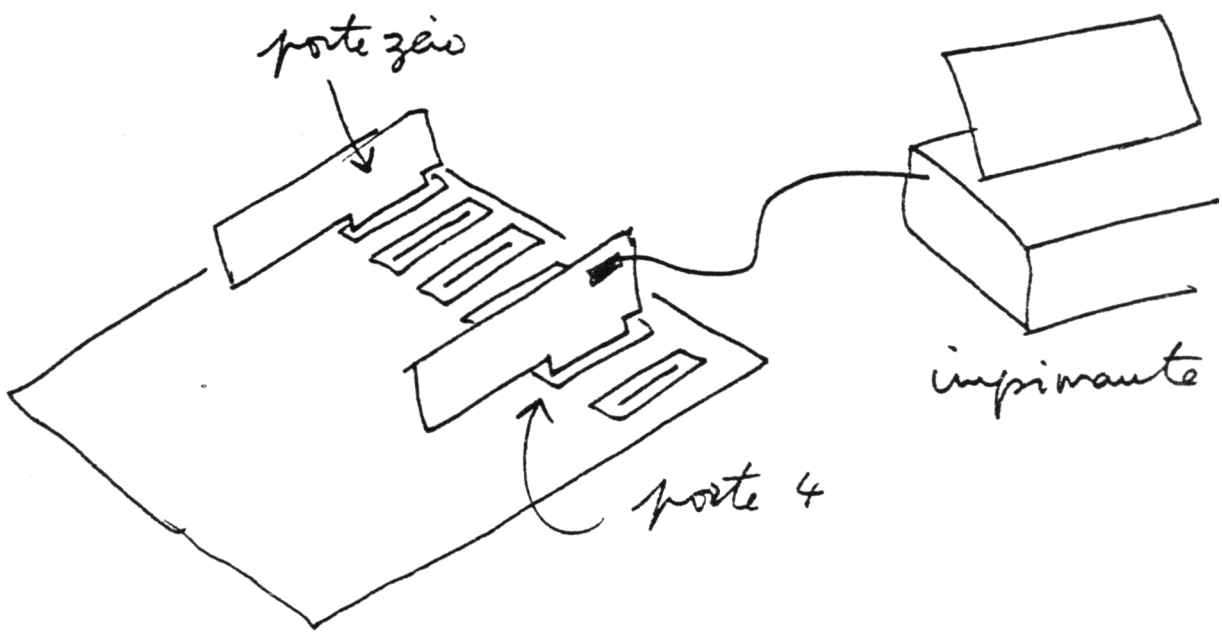
L'option 5 se réfère à la relance des impayés.

On va encore une fois explorer le fichier. Et rechercher les clients débiteurs. En ce cas on donnera les renseignements concernant ce compte.

Avant d'écrire ce sous programme, nous allons suggérer un exercice de "BUREAUTIQUE" que vous ferez vous même.
Vous écrirez un autre sous programme, qui sera appelé en cas de compte négatif (débit). Ce sous programme (mettez le par exemple à l'adresse 12ØØØ rédigera automatiquement la lettre de relance. Si on avait sur l'ORIC une imprimante, celle ci serait activée dans ce sous programme, et la lettre de relance sortirait. On aurait plus qu'à la glisser dans l'enveloppe.

Quand nous équipperons les ORIC, ou au moins l'un d'eux, avec une imprimante, il y aura une ordre d'activation, et un ordre de désactivation. Pour le moment nous ne les connaissons pas. Mais, à titre indicatif, nous indiquerons comment cela se passe avec un APPLE II.

Un APPLE II possède un certain nombre de "portes"



Ces "portes" s'appellent aussi "slots",ce qui veut dire fentes. Dans ces connecteurs d'un type particulier on peut disposer des CARTES ,qui peuvent être très variées.

Dans l'APPLE II on a sept slots,numérotés de gauche à droite de Ø à 7.Le slot sept est réservé à la sortie graphique-couleur, quand il y en a une (auquel cas dans ce slot se trouve une carte-couleur,elle même connectée à un téléviseur ou à un moniteur couleur à l'aide d'une prise PERITEL).

Le slot zéro est réservé aux "cartes langage".

Sur le dessin précédent on a reprsénté l'imprimante connectée à une carte enfichée par exemple dans le SLOT 4.

L'imprimante est un PERIPHERIQUE.

L'ordre PR # 4 va "activer" le périphérique correspondant au "slot" numéro quatre.

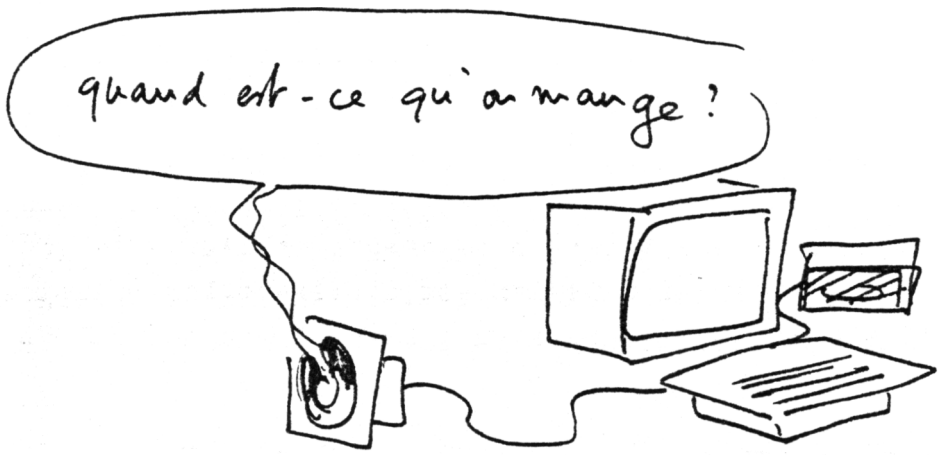
Si cet ordre est inclu dans le programme,tous les ordres PRINT se doubleront d'une impression sur le papier.

Si on veut faire cesser cette impression,il faudra consigner dans le programme l'odre PR # Ø qui rebranche l'APPLE II sur le slot zéro,c'est à dire sur sa "carte langage".

Sur l'ORIC il y aura quelquechose du même tonneau.Mais nous ne connaissons pas encore les ordres ad hoc.L'imprimente se branchera sur l'une des prises multibroches situées au dos de l'appareil.On en distingue deux.L'une sera vraisemblablement affectée au lecteur de diskette et l'autre à un périphérique comme par exemple une imprimante.

Ceci n'est qu'une parenthèse.Quand vous écrirez votre sous programme 12ØØØ vous ne tiendrez pas compte de cette possibilité d'impression.Mais en sachant que cela peut exister par la suite. Ceci dit,si le périphérique était un système de synthèse de parole, en activant ce périphérique,la machine pourrait alors interpréter les ordres PRINT comme des mots,et se mettrait à les énoncer à haute voix....

par exemple....



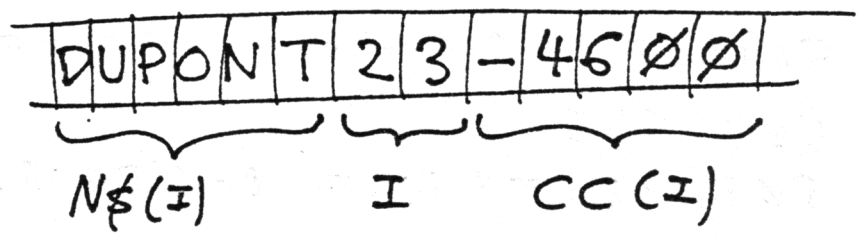
Revenons à cette séquence de relance des impayés :

```
5000 REM RELANCE IMPAYES
5010 CLS
5020 PRINT" SONT DEBITEURS :": PRINT
5030 FOR I = 1 TO NC
5040 IF CC(I)<0 THEN PRINT N$(I),
     I , CC(I): GOSUB 12000 : PRINT
5050 NEXT I
5060 PRINT"APPUYEZ SUR LA BARRE
D'ESPACEMENT
5070 GET A$
5999 RETURN
```

Ici nouvelle exploration du fichier. Mais il ne s'agit plus de chercher quelqu'un. Il faut explorer TOUT le fichier. Nous pouvons alors utiliser une boucle FOR NEXT.
Si le compte est en négatif (test) la machine donnera le nom du client, son numéro de compte et l'état de son compte CC(I). Mais l'ordre d'impression est différent. Les registre sont séparés non plus par des points virgule, mais par des VIRGULES.

```
PRINT N$(I);I;CC(I)
```

mettrait ces objets BOUT A BOUT :




```
PRINT N$(I),I,CC(I)
```

permet une sortie sur trois colonnes :

DUPONT	23	-46ØØ
DURAND	17	-1746

C'est une possibilité offerte par l'ORIC (et par la plupart des micros). La capacité est limitée à trois colonnes.

```
PRINT Q$,T
```

Donnerait deux objets (chaines ou nombres) sur deux colonnes.

Sur l'APPLE II cette capacité est limitée à trois colonnes. Mais sur l'ORIC il semble bien que l'on puisse aller plus loin.

A la ligne 5Ø6Ø vous trouvez un ordre print et en 5Ø7Ø une instruction nouvelle : GET

Cette instruction GET ressemble à l'instruction INPUT. Mais elle limite la saisie à UN SEUL CARACTERE. Qui sera ici placé dans le registre A$.
Dans l'exécution, le cours du programme est interrompu, et il y a attente de ce caractère. Vous pouvez alors presser sur d'importe quelle touche. La plus proche et la plus simple est la barre d'espacement. Mais vous pourriez aussi bien appuyer sur la touche Return.
Le caractère " " correspondant à la barre est alors stocké dans A$.
A quoi ça sert ?
A rien... sauf que cela permet d'interrompre le cours du programme pendant le temps que l'on veut (à la différence d'un WAIT). Pour faire repartir il suffit d'appuyer sur une touche. C'est un gadget très commode.

Nous allons continuer par une analyse des comptes-clients.

```
6ØØØ REM ANALYSE COMPTES-CLIENTS
6Ø1Ø CLS
6Ø2Ø PRINT"NOMBRE DE CLIENTS";NC
6Ø3Ø GET A$
6Ø4Ø CLS
```

```
6050 PRINT"VOULEZ-VOUS : ":PRINT
6060 PRINT"1-ACTIF & PASSIF":PRINT
6070 PRINT"2- ..... " :PRINT
6080 PRINT"3-.-.-.- " :PRINT
6090 PRINT"4-.-.- " :PRINT
6100 INPUT"VOTRE CHOIX";C
6110 ON C GOSUB 13000,14000,15000,16000
6999 RETURN
```

Je vous rappelée que vous avez toujours à écrire le programme

```
12000 REM LETTRE DE RELANCE

12999 RETURN
```

Le sous programme 6000 propose un sous menu. Nous ne décrirons que la première option.

```
12000 REM ANALYSE ACTIF/PASSIF
12010  A=0 : P=0
12020  FOR I = 1 TO NC

12030   IF CC(I)   0 THEN A = A + CC(I)

12040  IF CC(I)    0 THEN  P = P + CC(I)
12050 NEXT I
12060  PRINT" MONTANT COMPTES CLIENTS:";A;" F":PRINT
12070 PRINT"ENSEMBLE DES IMPAYES :";P;" F"
12080 GET A$
12999 RETURN
```

Une sconde opfion pourrait-être :

2- CLIENTS HABITANT LA FRANCE

Il s'agit alors d'écrire une sous routine d'exploration du fichier, à travers un FOR NEXT,très simple.Et on signale les noms et adresses des gens résidant en France.

On pourrait tout aussi bien sélectionner des gens habitant la France,ou un autre pays et dont les comptes sont positifs ou négatifs.

Tout ceci ne représente que des bribes de ce vaste problème qu'est la gestion.Il manque beaucoup d'éléments par ailleurs.
Dans la seconde partie du cours,on évoquera le problème du TRI.

Pour finir,nous allons faire le tour de ces éléments de notre MECCANO INFORMATIQUE que nous avons déballé.
Il en reste d'autres.

HARD & SOFT	p. 8
CLS	p. 16
PAPER INK	p. 17
REGISTRES	p. 18
PRINT ?	p. 20
CLEAR	p. 20
CHAINES DE CARACTERES N$	p. 22
PROGRAMME	p. 24
RUN	p. 27
NEW	p. 28
LIST	p. 28
LIST -30 LIST 20- LIST 30-50	p. 29
END	p30

Inégalités
<
<=
>
>=
<>
AND
OR p. 50
p. 51

SAUVER un Programme p. 54
SAVE
LOAD
ALGORITHME p. 57
INCREMENTATION
INCREMENTATION p. 62

BOUCLES p. 70

STEP p. 72

Control C p. 73

WAIT p. 79

INT p. 80